Microsoft SQL documentation

Learn how to use SQL Server and Azure SQL, both on-premises and in the cloud.

DOWNLOAD
Try SQL Server 2022

OVERVIEW
Get an Azure VM with SQL Server

OVERVIEW
SQL tools

REFERENCE
Troubleshoot

SQL Server on Windows

On-premises

On an Azure VM

Previous versions (SQL 2005-2014)

SQL on Azure

Azure SQL Database

Azure SQL Managed Instance

Azure SQL Edge

Azure Synapse Analytics

SQL Server on Linux

On-premises

On an Azure VM

Getting started

- What is SQL Server?
- Connect to the Database Engine
- Connect and query with SSMS
- Write Transact-SQL
- SQL Server in a Linux container
- SQL Docs navigation tips

What's new

- SQL Server 2022
- Azure SQL Database
- Azure SQL Managed Instance
- SQL Server on Azure VMs
- Azure Data Studio
- SQL Server Machine Learning Services

Educational SQL resources

SQL Server enabled by Azure Arc

Install

- SQL Server
- SQL Server Management Studio (SSMS)
- Azure Data Studio
- Analysis Services (SSAS)
- Integration Services (SSIS)
- Reporting Services (SSRS)

Migrate

- Data migration
- Database compatibility certification
- Azure Database Migration Service (DMS)
- Database Migration Assistant (DMA)
- SQL Server Migration Assistant (SSMA)

See more >

Development

- SQL Server client connectivity
- Azure SQL Database applications
- Azure SQL Database connectivity libraries
- Transact-SQL (T-SQL)
- SQL PowerShell
- Database samples

Reporting and analysis

- Power BI Report Server
- Power BI Service
- SQL Server Reporting Services (SSRS)
- SQL Server Analysis Services (SSAS)
- Azure Analysis Services

Big data and machine learning

- Azure Synapse Analytics
- SQL Server Machine Learning Services (R & Python)
- SQL Server Big Data Clusters
- Data Virtualization with PolyBase
- Microsoft Analytics Platform System

SQL Server Management Studio 21 Preview FAQ

FAQ

SQL Server Management Studio 21 Preview is the latest preview version of Microsoft's integrated environment for managing any SQL infrastructure, from SQL Server to Azure SQL.

What is SQL Server Management Studio 21 Preview?

SQL Server Management Studio 21 Preview is the latest preview version of Microsoft's integrated environment for managing any SQL infrastructure, from SQL Server to Azure SQL. SQL Server Management Studio (SSMS) provides the ability to configure, monitor, and administer instances of SQL Server, as well as Azure SQL Managed Instance and Azure SQL databases. Use SSMS to deploy, monitor, and upgrade the data-tier components used by your applications, as well as develop queries and scripts. SSMS runs on Windows and is a free download from Microsoft.

What are the new features in SSMS 21?

SSMS 21 includes several new features and enhancements:

- **Update to Visual Studio 2022**: SSMS 21 is now based on Visual Studio 2022.
- **64-bit Architecture**: SSMS is now a 64-bit application, ensuring smoother operations and reduced out-of-memory errors.
- **Enhanced User Interface**: A refreshed, modernized design with updated icons, the initial phase of dark theme, and customizable layouts.
- **Enhanced Query Editing**: New options exist within the query editor to improve efficiency when editing and managing scripts.
- **Improved Azure Administration**: Improved support for managing users and logins for Azure SQL Database.
- **Compatibility**: Full compatibility with SQL Server 2022, including support for the latest features.
- **Git Integration**: Integration with Git for tracking, managing, and collaborating on SQL scripts.
- **Performance & Reliability**: Numerous under-the-hood enhancements for smoother performance and improved stability.

- **Security Features**: A new Always Encryption Assessment wizard, and an updated Azure Authentication experience.

For a full list of new features, visit the official release notes.

Where can I find more information about SSMS 21?

For more information about SSMS 21, including release notes and documentation, visit the official Microsoft Learn content for SQL Server Management Studio 21.

Is SSMS 21 compatible with previous versions of SQL Server?

Yes, SSMS 21 is designed to be backward compatible with previous versions of SQL Server. You can use it to manage all supported SQL Server versions, such as SQL Server 2022, SQL Server 2019, and earlier versions.

Can I use SSMS 21 to manage Azure SQL Database?

Yes, SSMS 21 includes enhanced integration for managing Azure SQL Database and Azure SQL Managed Instance environments.

What are the system requirements for SSMS 21?

The system requirements for SSMS 21 include:

- **Operating System**: Windows 10 or later
- **Processor**: 1.8 GHz or faster
- **Memory**: 4-GB RAM minimum
- **Hard Disk Space**: 4 GB of available disk space
- **Display**: 1280 x 800 resolution or higher

Does SSMS 21 run on macOS?

No. SSMS is available only for Windows.

How do I install SSMS 21?

To install SSMS 21, you must use the Visual Studio installer. Follow the on-screen instructions to complete the installation process.

Do I have to install Visual Studio to use SSMS 21?

No. SSMS 21 doesn't require any edition of Visual Studio (for example, Community, Enterprise) to be installed. SSMS 21 isn't an extension or workload of Visual Studio.

What if I already have Visual Studio 2022 (or Visual Studio 2019) installed?

It doesn't matter! SSMS 21 is installed in a separate location that you can select in the installer.

Can I install Visual Studio 2022 after I've installed SSMS 21?

Yes.

How much does SSMS 21 cost?

SSMS 21 is free for personal or enterprise use.

Do I need a Visual Studio subscription to install or use SSMS 21?

No. SQL Server Management Studio 21 Preview has its own EULA) and remains a free application, just as it always has been.

Who should use SSMS 21?

Anyone can use SSMS 21! However, it's designed to simplify tasks performed by database administrators, database developers, system administrators, and independent software vendors.

What can I do with SSMS 21?

SSMS 21 is a Visual Studio 2022-based solution, and offers an integrated experience when working with SQL Server, Azure SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytics. SSMS 21 makes the core experiences that you rely on every day simple and easy with built-in features such as multiple tab windows, a rich SQL query editor, IntelliSense, keyword completion, code snippets, and source control integration (Git). You can execute queries on-demand, view, and save results as text or CSV, edit data, and browse database objects in object explorer.

SSMS 21 offers a consistent experience to back up and restore your databases. With support for SQL Server Always On Availability Groups, you can easily configure, monitor, and troubleshoot Availability Groups for your mission-critical SQL Server databases and quickly failover to a secondary database during a disaster. SSMS has been designed to make you more productive in the DevOps lifecycle of your databases of choice on the operating systems of your choice. As a result, you're always in control, and you can reduce risks, solve problems faster, and continuously deliver value that exceeds customers' expectations.

Is SSMS 21 Open Source?

No.

Do you plan to open-source SSMS 21?

No.

When should I use SSMS 21?

Use SQL Server Management Studio 21 if you:

- Are doing complex administrative or platform configuration.
- Are doing security management, including login, user, and role management, and implementation of Always Encrypted.
- Need to make use of performance tuning advisors and dashboards.
- Use database diagrams and table designers.

- Need access to Registered Servers.
- Make use of live query stats or client statistics.

Who typically uses SSMS?

Database administrators (DBAs) typically use SSMS, but any data professional such as database administrator or system administrator can use SSMS.

Does SSMS 21 integrate with Azure Synapse Analytics?

Yes.

Does SSMS 21 integrate with Fabric DW?

You can connect to a Fabric DW database from SSMS 21, but there are numerous improvements that need to be made in order to provide full support.

Why isn't dark theme available for every window?

We'll make dark theme available across more dialogs in SSMS 21 over time. Initial support exists for dialogs that are most frequently accessed by users, including Object Explorer, Query Editor, Results pane, and Template Explorer. If you plan to use dark theme, and would like to provide feedback on the dialogs that you use the most, please go to the SSMS 21 dark theme support feedback item and add a comment.

Where is Copilot for SSMS?

Copilot for SSMS will soon be available in private preview.

Will Copilot for SSMS be installed by default?

No. When available, Copilot for SSMS will be an opt-in experience, available as an extension.

Will there be additional previews for SSMS 21?

Yes, there will be multiple previews of SSMS 21. Notifications will occur within SSMS 21 when a new preview is available.

Where do I report issues or feedback requests for SSMS 21?

Use the SSMS Developer Community site.

Feedback

Was this page helpful? Yes No

Release notes for SQL Server Management Studio 21 Preview

Article • 03/11/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

This article details updates, improvements, and bug fixes for the current and previous versions of SQL Server Management Studio 21 Preview.

> **Note**
>
> In December 2021, releases of SSMS prior to 18.6 will no longer authenticate to Database Engines through Microsoft Entra multifactor authentication (MFA). To continue utilizing MFA, you need SSMS 18.6 or later.
>
> Connectivity to Azure Analysis Services through Microsoft Entra multifactor authentication requires SSMS 18.5.1 or later.

Current SQL Server Management Studio 21 preview release

Download SQL Server Management Studio 21 Preview

SQL Server Management Studio 21 Preview is the latest preview release of SSMS. If you need a previous version of SSMS, see previous SSMS releases.

21.0 Preview 5

- Release number: 21.0 Preview 5
- Build number: 21.0.107
- Release date: March 11, 2025

What's new in 21.0 Preview 5

Expand table

Feature	Details
Azure SQL Database	Added support for MANUAL_CUTOVER and PERFORM_CUTOVER. For more information, see Improving the conversion to Hyperscale with greater efficiency.
Command Line	Added the `-A` option to specify the authentication method to use when connecting from the command line. Accepted values are authentication types from SqlAuthenticationMethod supported by SSMS. The `-G` (use Entra ID authentication) and `-E` (use Windows Authentication) options are now deprecated and will be removed in a later version.
Connection	When you connect to a server from command line parameters, SSMS displays a prompt to confirm the connection and parameters.
Connection	Added **Close Idle SQL Connections** command under the Help menu. For more information, see Connection Pooling in SQL Server Management Studio.
IntelliSense	Added support for Vector functions.
Individual Components (Visual Studio Installer)	Reinstated Analysis Services and Reporting Services as Business Intelligence components in the Visual Studio Installer.
Libraries	Updated Server Management Objects (SMO) to version 17.100.64.
Scripting	Added support for scripting objects that use the Vector data type.
Theming	Improved dark theme support for the Activity Monitor and Object Explorer Details dialogs.
Tools	Removed Azure Data Studio entry from the Tools menu. For more information, see What's Happening to Azure Data Studio.
Visual Studio	Updated to Visual Studio 17.14 Preview 2.

Bug fixes in 21.0 Preview 5

Expand table

Feature	Details
Accessibility	Fixed screen reader compatibility and keyboard navigation issues in the Always Encrypted wizard.
Accessibility	Fixed tabbing order for the **Replace in Files** dialog. See In SSMS 21.0 Replace in Files the tab order is incorrect.

Feature	Details
Accessibility	Fixed keyboard navigation issues in Profiler.
File	Addressed an issue that created duplicate entries in the **Add New Item** dialog when configuring a Project in the Solution Explorer.
Find	Resolved issue with **Find in Files** when searching in text results, accessed through **Edit** > **Find and Replace** > **Find in Files**, or with CTRL + F. See Support find in SSMS 21 text results window .
Import/Export Data-Tier Application	Updated the file and log location to use default instead of master database location during a bacpac import.
Object Explorer	Addressed an issue that incorrectly displayed Vector data type columns.
Query Editor	Resolved an issue where characters in the range of 160-191 were displayed with a strikethrough.

21.0 Preview 4.1

- Release number: 21.0 Preview 4.1
- Build number: 21.0.85
- Release date: February 24, 2025

What's new in 21.0 Preview 4.1

Expand table

Feature	Details
Visual Studio	Updated to Visual Studio 17.14 Preview 1.1.

21.0 Preview 4

- Release number: 21.0 Preview 4
- Build number: 21.0.85
- Release date: February 11, 2025

What's new in 21.0 Preview 4

Expand table

Feature	Details
Account Authentication	Added theme support for Account picker.
File	Reinstated the ability to create projects (**File** > **New** > **Project or Solution...**) and view recent projects (**File** > **Open** > **Project/Solution...**).
Help	Enabled support for Developer Community feedback, accessed from **Help** > **Send Feedback** > **Report a Problem/Suggest a Feature/My Feedback**.
IntelliSense	Added support for `JSON_CONTAINS` built-in functions.
IntelliSense	Added support for `REGEXP_` functions. See SSMS 21 Preview 2: REGEXP_ functions need to be color-coded.
Query Store	Added theme support for the search window and grid in the Tracked Queries report.
Visual Studio	Updated to Visual Studio 17.14 Preview 1.

Bug fixes in 21.0 Preview 4

Expand table

Feature	Details
Options	Removed localization of `ISOLATION LEVEL` option within **Tools** > **Options** > **Query Execution** > **SQL Server** > **Advanced**.
IntelliSense	Corrected default IntelliSense toolbar icon state to be disabled when IntelliSense is disabled globally.
Query Plans	Addressed an issue where canceling a query while running a live query plan caused SSMS to crash. See Live query statistics hangs waiting for plan and cancelling query kills SSMS.
Query Plans	Addressed an issue where viewing live query statistics for a table with a geography data type caused an error.

21.0 Preview 3

- Release number: 21.0 Preview 3
- Build number: 21.0.83
- Release date: January 22, 2025

What's new in 21.0 Preview 3

Expand table

Feature	Details
Always Encrypted	Introduced online encryption support in the Always Encrypted Wizard, which enables incremental copying, encrypting, decrypting, or re-encrypting of data. See Configure column encryption using Always Encrypted Wizard in SSMS 21 Preview.
Azure SQL Managed Instance	Introduced new validation steps in the Failover Readiness Assessment dialog to assess the health status of the link before failover.
Configuration	The default value for a firewall rule name created in SSMS uses the format `ClientIPAddress_YYYY-MM-DD_HH-MM-SS`.
Find	Reinstated the **Quick Find** option, accessed through **Edit** > **Find and Replace** > **Quick Find**, or with **CTRL** + **F**. See SSMS 21.0 Preview 1.0 - "Quick Find" missing, instead "Find in files" pops up.
Installation	Updated the default Code Snippets folder location to be major-version specific (`%USERPROFILE%\Documents\SQL Server Management Studio 21\Code Snippets\SQL\My Code Snippets`).
Query Store	Added theme support for Plan ID box and toolbar dropdown lists in Query Store reports.
Intellisense	Added support for VECTOR data type.
Visual Studio	Updated to Visual Studio 17.13 Preview 3.

Bug fixes in 21.0 Preview 3

Expand table

Feature	Details
Always Encrypted	Addressed an issue where the default constraint was removed after performing in-place encryption in the Always Encrypted Wizard.
Connection	Updated the database Connection Properties dialog to show the correct authentication method.
Intellisense	Disabling IntelliSense by unchecking **Enable IntelliSense** in **Tools** > **Options** > **Text Editor** > **Transact-SQL** > **IntelliSense** now persists between restarts of SSMS. See Intellisense enabled for SSMS 21 despite being off in the settings.

Feature	Details
License	Updated the "Microsoft Software License Terms" link in the Visual Studio Install splash screen to direct to the SSMS 21 license.
Query Plans	Updated operator Property dialog to correctly display Columns With Stale Statistics warning.
Query Store	Improved readability for Plan ID text in the Top Resource Consuming Queries report for High Contrast Mode.
Security	Added the ability to assign a new user to a database login from the user properties dialog.
Security	Corrected size of "Add" button on the Login Properties dialog.
SQL Agent	Removed unsupported SQL Agent features for Azure SQL Managed Instance including targets, multi-server administration, and Operator pager options.

21.0 Preview 2.1

- Release number: 21.0 Preview 2.1
- Build number: 21.0.73
- Release date: December 18, 2024

What's new in 21.0 Preview 2.1

Expand table

Feature	Details
Visual Studio	Updated to Visual Studio 17.13 Preview 2.1.

21.0 Preview 2

- Release number: 21.0 Preview 2
- Build number: 21.0.73
- Release date: December 10, 2024

What's new in 21.0 Preview 2

Expand table

Feature	Details
Accessibility	Improved accessibility in the View Designer.
Always Encrypted	Introduced the capability to generate new keys within the Always Encrypted Wizard that are securely stored in Azure Key Vault Managed HSM (Hardware Security Model).
Always Encrypted	Improved performance of the New Column Master Key dialog.
Azure Authentication	Introduced the ability to select a tenant within an Azure subscription. Tenant selection is available in dialogs such as creating a new firewall rule, using the Azure Storage browser, and selecting a container for backup to, or restore from, URL.
Connection	Updated strict encryption label to **Strict (Minimum SQL 2022 or Azure SQL)** in the **Encryption** dropdown list on the connection dialog.
Query Editor	Intellisense errors are updated to map to the IntelliSense dropdown list in the Error List pane. The Error List pane is available from the **View** > **Error List** menu or by selecting the error icon in the file health indicator, located in the bottom left of the Query Editor toolbar.
Settings	Changed default startup option for SSMS back to **Open Object Explorer**, within **Tools** > **Options** > **Startup**.
Settings	Removed the **Preview** entry from the **Tools** > **Options Experience (requires restart)** menu, as SSMS doesn't currently support Preview Unified settings.
Theming	Added dark theme support to tabs and icons in the Results pane of the Query Editor, and to the script splitter bar.
Theming	Added dark theme support to the query and plan panes within Query Store reports.
Visual Studio	Updated to Visual Studio 17.13 Preview 2.

Bug fixes in 21.0 Preview 2

Expand table

Feature	Details
Always Encrypted	Renamed **Encrypt Column** to **Always Encrypted Wizard** in the Object Explorer context menu.
Always Encrypted	Introduced status **Completed With Warnings** for the Always Encrypted Wizard.

Feature	Details
Connection	Resolved inability to reconnect a query editor after a forced disconnection. See Forcefully disconnect query window can't reconnect.
Connection	Updated check for open transaction count to not execute against SQL offerings that don't support transactions (for example, Azure synapse serverless pools). See Query transaction count should not run if enabled from SSMS and underlying engine doesn't support transactions.
Connection	Resolved issue where connection dialog might appear while SSMS is still loading.
Database Mirroring	Resolved issues in Database Mirroring Monitor where mirrored databases couldn't be registered and server instance connections can't be added.
Installer	Corrected the product version listed on the Details page within the Properties of the bootstrapper `vs_SSMS.exe`.
Installer	Updated installation to place SSMS 21 and related applications in a folder named **Microsoft SQL Server Tools 21**. See Rename SSMS 21 to align with pervious versions in Control Panel.
Object Explorer	Resolved error "Couldn't load file or assembly 'Microsoft.DataWarehouse, Version=16.2.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91' or one of its dependencies. The system can't find the file specified." when trying to filter in Object Explorer. See SSMS 21 Preview 1 - Right-click "Filter settings" does not work.
Object Explorer	Removed the overlapped display new object name and original object name when renaming an object in Object Explorer.
Query Plans	Resolved behavior where selecting showplan XML in the results grid opened the plan as XML instead of the graphical display.
Query Store Reports	Fixed problem where changes to configuration in the Tracked Queries report weren't retained.
Query Store Reports	Implemented sizing restrictions for the Configuration dialog to prevent a degraded display.
Scripting	Addressed a problem where the updated value for **Auto Create Incremental Statistics** wasn't included when scripting the change for Azure SQL Database.
Scripting	Removed `USE Database Name` syntax when generating scripts for users in Azure SQL Database.
Settings	Fixed issue where importing settings on a non-English Operating System (OS) generated an error.
Settings	Removed **CodeLens** from **Tools** > **Options** > **Text Editor** > **All Languages**.

Feature	Details
Settings	Removed duplicate and irrelevant entries from **Tools** > **Options** > **Text Editor** > **File Extensions**.
Theming	Resolved an issue where the Messages tab in the Results pane wouldn't pick up a theme change until SSMS was restarted.
Theming	Improved handling of NULL cells in the results grid.
Theming	Fixed incorrect theming for Extended Events context menus.
Theming	Improved themed support for **Recently Used Templates** in Template Explorer.
Theming	Improved theme support in the Messages tab of the results pane.
Theming	Addressed issue where the results window would show as white until the results appeared with dark theme enabled.

21.0 Preview 1

- Release number: 21.0 Preview 1
- Build number: 21.0.47
- Release date: November 12, 2024

What's new in 21.0 Preview 1

Expand table

Feature	Details
64-bit	SSMS 21 is a Visual Studio 2022-based application that supports 64-bit, ensuring smoother operations and reducing out-of-memory errors.
Always Encrypted	An assessment for Always Encrypted is available in the wizard. It can be found under the **Always Encrypted Wizard** menu (formerly named **Encrypt Columns...**) after right-clicking a database and selecting **Tasks**. For more information, see Always Encrypted Assessment in SQL Server Management Studio 21.
Azure Authentication	Azure Authentication encompasses a subset of dialogs where authentication to Azure is required. Examples of these dialogs include adding an Azure firewall rule, configuring the Service Level Objective (SLO) when creating an Azure database, and the Azure storage browser.
Azure SQL Database	Introduced UI support for creating logins and database users.

Feature	Details
Dark theme	SSMS 21 supports dark theme for selected dialogs, including Object Explorer, Query Editor, Results pane, and Template Explorer.
Database Properties	Added a page to list Database Scoped Configuration values on the **Database Properties** dialog.
Drivers	Updated Microsoft.Data.SqlClient (MDS) to version 5.1.6.
Git	Support for Git integration returns in SSMS 21 to support the growing emphasis on CI/CD, which can be found in the **Git** menu. Select **Settings** to configure options specific to Git. The Git integration allows developers and database administrators to track, manage, and collaborate on SQL scripts seamlessly within the SSMS environment, including support for creating and cloning repositories and creating and managing branches.
Installation	SSMS 21 is installed using the Visual Studio Installer. For more information, see Install SQL Server Management Studio 21 Preview.
Language support	Introduced support for SSMS in Czech, Polish, and Turkish.
Libraries	Updated Azure.Core to version 1.41.0.
Libraries	Updated DacFx to version 162.4.92.
Libraries	Updated Server Management Objects (SMO) to version 17.100.52.
Libraries	Updated System.Text.Json to version 8.0.4.
Query Editor	The scroll bar for the query editor in SSMS 21 defaults to **map mode**. To change the option, right-click the scroll bar and select **Scroll Bar Options...**. Within the **Options** dialog, you can change the display to **Use bar mode for vertical scroll bar**.
Query Store	The Query Store reports feature a **Replica** dropdown list, allowing users to view Query Store data across various replica sets or roles. For more information, see sys.query_store_replicas to understand the current association between a replica and its role. When the Query Store for secondary replicas feature is enabled, data populates exclusively for roles that are deployed and designated as replicas.
Sign in	Users can now access their Azure and GitHub accounts from SSMS 21. Sign-in isn't required to install or use SSMS 21. For more information, see Access multiple accounts in SQL Server Management Studio 21 Preview.
Terminal	Introduced integrated terminal access from the **View** menu to support writing and executing command line and PowerShell commands.
User Interface	The updated SSMS interface has a refreshed, modernized design optimized for a streamlined experience that integrates with high-DPI displays. The new design

Feature	Details
	includes updated icons, a modern dark theme, and customizable layouts to suit your preferences.
Visual Studio	Updated to Visual Studio 17.13 Preview 1.

Bug fixes in 21.0 Preview 1

Expand table

Feature	Details
Azure SQL Database	Addressed issue of SSMS hanging when trying to connect to an Azure SQL Database that was deleted.
Azure SQL Database	Resolved the problem of SSMS becoming inaccessible when a user doesn't have permissions to access all the databases on a logical server.
Azure SQL Database	Fixed the behavior of SSMS hanging and generating the error "An error occurred while changing the current database." The error occurs when an invalid database name is entered in the database name dropdown list for the editor window.
Azure SQL Managed Instance	Removed the ability to select an alternate option for **Login auditing** within the **Server Properties** dialog.
Azure SQL Managed Instance	Removed the ability to configure the number of error log files in the **Configure SQL Server Error Logs** dialog.
Central Management Server	Added ability to save the **Trust Server Certificate** connection option.

Known issues 21.0 Preview

See Known issues in SQL Server Management Studio 21 Preview for known issues using SQL Server Management Studio 21 Preview.

Support or troubleshooting

Sometimes, things can go wrong. If your SSMS installation fails, see Troubleshoot installation and upgrade issues for SQL Server Management Studio 21 Preview for step-by-step guidance.

You can reference the SSMS Developer Community site for other known issues (filter on **Tooling** under **Groups**) and to provide feedback to the product team.

Related content

- Install SQL Server Management Studio 21 Preview
- SQL Server Management Studio 21 Preview FAQ
- Visual Studio 2022 Preview release notes

Feedback

Was this page helpful? Yes No

System requirements for SQL Server Management Studio 21 Preview

Article • 02/13/2025

SQL Server Management Studio 21 Preview is a powerful tool for managing SQL Server instances, Azure SQL databases, and Azure SQL managed instances. This document outlines the system requirements needed to install and run SQL Server Management Studio 21 Preview effectively.

Ensure your system meets these requirements to take full advantage of the features and capabilities of SQL Server Management Studio (SSMS). For more information about SQL Server Management Studio 21 Preview, see Release notes for SQL Server Management Studio 21 Preview.

Download SQL Server Management Studio 21

To download the latest version of SQL Server Management Studio 21 Preview, see Install SQL Server Management Studio 21 Preview. For instructions on installing and updating SQL Server Management Studio 21 Preview, see Update SQL Server Management Studio 21 Preview. Also, see instructions on how to install offline.

Download SSMS 21 Preview

> **Tip**
>
> For SQL Server Management Studio 20 and earlier versions, see **Download SQL Server Management Studio (SSMS)**.

SQL Server Management Studio 21 system requirements

SQL Server Management Studio 21 Preview is supported on the following 64-bit operating systems:

- Windows 11 minimum supported OS version or higher: Home, Pro, Pro Education, Pro for Workstations, Enterprise, and Education
  - For supported Windows 11 versions, see Windows 11 Enterprise and Education Support

- Windows 10 minimum supported OS version or higher: Home, Professional, Education, and Enterprise.
  - For supported Windows 10 versions, see Windows 10 Enterprise and Education Support
- Windows Server 2022: Standard and Datacenter.
- Windows Server 2019: Standard and Datacenter.
- Windows Server 2016: Standard and Datacenter.

The following operating systems aren't supported:

- 32-bit and ARM32 operating systems.
- Windows 11 Home in S mode, Windows Enterprise IoT, Windows 10 IoT Core, Windows 10 Enterprise LTSC edition, Windows 10 S, and Windows 10 Team Edition.
- Server IoT and Minimal Server Interface options for Windows Server.
- Windows containers.
- Running in virtual machine environment without a full Windows operating system.
- Application virtualization solutions such as Microsoft App-V or MSIX for Windows, or third-party app virtualization technologies.
- Multiple simultaneous users using the software on the same machine, including shared virtual desktop infrastructure machines or a pooled Windows Virtual Desktop hostpool.

Hardware

- x64 processor; Quad-core or better recommended.
  - ARM64 processors aren't supported for SSMS 21 Preview 1
  - ARM32 processors aren't supported.
- Minimum of 4-GB of RAM. Many factors impact resources used; we recommend 16-GB RAM for typical professional solutions.
- Hard disk space: Minimum of 4-GB up to 50-GB of available space, depending on features installed; typical installations require 20-50 GB of free space. We recommend installing Windows and SSMS on a solid-state drive (SSD) to increase performance.
- Video card that supports a minimum display resolution of WXGA (1366 by 768); SSMS works best at a resolution of 1920 by 1080 or higher.
  - Minimum resolution assumes zoom, DPI settings, and text scaling are set at 100%. If not set to 100%, minimum resolution should be scaled accordingly. For example, if you set the Windows display **Scale and layout** setting on your Surface Book, which has a 3000x2000 physical display, to 200%, then SSMS

would see a logical screen resolution of 1500x1000, meeting the minimum 1366x768 requirement.

Supported languages

SQL Server Management Studio 21 Preview is available in English, Chinese (Simplified), Chinese (Traditional), Czech, French, German, Italian, Japanese, Korean, Polish, Portuguese (Brazil), Russian, Spanish, and Turkish. You can select the language of SSMS during installation. The Visual Studio Installer is available in the same 14 languages, and matches the language of Windows, if available.

Additional requirements and guidance

- Administrator rights are required to install or update SQL Server Management Studio 21 Preview.
- Refer to Create an offline installation of SQL Server Management Studio 21 Preview for guidance on how to install, deploy, update, and configure SQL Server Management Studio in an offline scenario.
- .NET Framework 4.7.2 or above is required to *install* SQL Server Management Studio 21 Preview. SQL Server Management Studio 21 Preview requires .NET Framework 4.8 to run. If .NET Framework 4.8 isn't already installed, it's installed during setup.
- Team Foundation Server 2019 Office Integration requires Office 2016, Office 2013, or Office 2010.
- Smart App Control, which is a Windows feature, isn't recommended to be enabled on machines. Any setting other than "off" might negatively impact SQL Server Management Studio performance.

Related content

- Install SQL Server Management Studio 21 Preview
- Modify SQL Server Management Studio 21 Preview components and language packs
- Create an offline installation of SQL Server Management Studio 21 Preview
- Use command-line parameters to install SQL Server Management Studio 21 Preview

---

Feedback

Was this page helpful? Yes No

Known issues in SQL Server Management Studio 21 Preview

Article • 03/11/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

This page lists known issues for SQL Server Management Studio 21 Preview.

Expand table

Feature	Details	Workaround
Arm64	There's no support for Arm64, and SSMS can't be installed on Arm64 devices.	Run SSMS on a device that isn't Arm64.
Database Diagrams	Closing a new database diagram before saving generates the error "Object reference not set to an instance of an object. (SqlEditors)".	Save the new database diagram before closing the editor in Database Diagrams.
Designers	Closing the view editor before saving in the View Designer generates the error "Object reference not set to an instance of an object. (SqlEditors)".	Save the new view before closing the editor in the View Designer.
Designers	The JSON data type isn't available for columns when creating or editing a table in Table Designer. See SSMS 21.0 Preview 2.0 - Table Designer missing the "json" datatype.	Use T-SQL to add columns with the JSON data type, this functionality is planned for a later release.
Designers	Unable to use SSMS after closing the **New View** pane when the **Add Table** dialog is left open. See SSMS 21 Preview 1: New View window "Add Table" not aware I closed out of New View.	Close the **Add Table** dialog before closing the view pane.
Extended Events	The Specify Data Storage page is truncated in the Extended Events Wizard. See Extended Events Wizard is cut off on Specify Data Storage page.	Use the **New Session...** option to create an Extended Events session.
Integration Services	There's currently no support for Integration Services in SQL Server Management Studio 21 Preview.	Use SSMS 20.2 to connect to Integration Services, we'll bring

Feature	Details	Workaround
		back Integration Services in a future preview.
Logins	Unable to map a credential for a login. See SSMS v21 Preview 1 - Bug in Login Properties for Credentials.	Use T-SQL to map a login to a credential.
Maintenance Plans	There's currently no support for creating or editing maintenance plans in SQL Server Management Studio 21 Preview.	Use SSMS 20.2 to create or edit maintenance plans, we'll bring back Integration Services in a future preview.
Menu	The **File** > **File** menu option isn't available.	Use **File** > **New Query with current connection** or the New Query button to open a new query editor.
Menu	Opening a folder from **File** > **Recent Projects and Solutions** generates one of the following errors: "System.InvalidOperationException: Can't enque project dependencies calculation before starting solution load" or "An exception of type NullReferenceException has been encountered." if opening the folder also opens one or more files that were open in the editor when the folder was last closed.	Closing the error allows work to continue. Alternatively, close all files in the editor before closing a folder.
Options	Enabling Per Monitor Awareness (PMA) by checking **Optimize rendering for screens with different pixel densities (requires restart)** within **Tools** > **Options** can cause issues with dialogs not rendering.	Don't enable **Optimize rendering for screens with different pixel densities (requires restart)** within **Tools** > **Options**.
Options	Enabling **Unified Settings Experience (requires restart)** within **Tools** > **Options** > **General** > **Preview Features** can cause issues with dialogs not rendering.	Don't enable **Unified Settings Experience (requires restart)** within **Tools** > **Options** > **General** > **Preview Features**.
Results Grid	When you switch themes, if Results Grid is set to Results to Text, the Results component doesn't switch themes, it applies the theme SSMS uses when it originally opens.	To apply any theme changes to the Results component, restart SSMS.

Feature	Details	Workaround
Settings	Changing font settings doesn't persist after a restart. See SSMS 21 Preview 1 - Environment Font Settings Not saving in a particular use-case.	Change the theme at the same time as the font settings.
Terminal	Using **Developer PowerShell** or **Developer Command Prompt** from **View** > **Terminal** generates an error. See SSMS 21 Preview - Visual Studio developer prompt options should probably be hidden or removed.	Customize the file that launches through customize what launches in SSMS, through **Tools** > **Options** > **Environment** > **Terminal**. A fix is expected in a future preview.

Related content

- Install SQL Server Management Studio 21 Preview

Feedback

Was this page helpful? Yes No

Install SQL Server Management Studio 21 Preview

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

Welcome to SQL Server Management Studio 21 Preview. In this version, it's easy to choose and install only the features you need, using the Visual Studio Installer.

> **Note**
>
> If you need to programmatically install SQL Server Management Studio, see Use command-line parameters.

Want to know more about what else is new in this version? See the release notes.

You can download and install a specific version from the release history page.

Ready to install? This article walks you through it, step-by-step.

Step 1 - Make sure your computer is ready for SQL Server Management Studio 21 Preview

Before you begin installing SQL Server Management Studio (SSMS):

- Check the system requirements. These requirements help you know whether your computer supports SQL Server Management Studio 21 Preview.
- Make sure that the user who performs the initial installation has administrator permissions on the machine. For more information, see User permissions and SQL Server Management Studio 21 Preview.
- Apply the latest Windows updates. These updates ensure that your computer has both the latest security updates and the required system components for SSMS.
- Restart. Restarting ensures that any pending installs or updates don't hinder your SSMS install.
- Free up space. Remove unneeded files and applications from your system drive by, for example, running the Disk Cleanup application.

You can install SQL Server Management Studio 21 Preview side-by-side with other versions. For more information, see Install SQL Server Management Studio versions side-by-side.

Step 2 - Determine which version of SQL Server Management Studio 21 to install

Decide which version of SSMS to install. The most common options are:

- The latest release of SQL Server Management Studio 21 Preview that is hosted on Microsoft servers. To install this version, select the following link. The installer downloads a small *bootstrapper* to your *Downloads* folder.

Download SSMS 21 Preview

- If you already have SQL Server Management Studio 21 installed, you can install another version alongside it.
- You can download a bootstrapper or installer for a specific version from the Release history page and use it to install another version of SSMS.

Step 3 - Initiate the installation

If you downloaded a bootstrapper file, you can use it to install SSMS. You need administrator permissions. The bootstrapper installs the latest version of the Visual Studio Installer. The installer is a separate program that provides everything you need to both install and customize SSMS.

1. From your *Downloads* folder, double-click the bootstrapper named `vs_ssms.exe` to start the installation.
2. If you receive a User Account Control notice, choose **Yes**. The dialog box asks you to acknowledge the Microsoft Privacy Statement and Microsoft Software License Terms. Choose **Continue**.

The Visual Studio Installer opens. In addition to SQL Server Management Studio, you can also install any product that Visual Studio Installer's **Available** tab offers.

Step 4 - Customize installation (optional)

After you install the Visual Studio Installer, you can use it to customize your installation SQL Server Management Studio by selecting different options on the tabs of the

installer. Here's how.

- SQL Server Management Studio 21 Preview 1 doesn't have any feature sets, or *workloads* available to install on the **Workloads** tab.
- You can add individual components to your installation from the **Individual components** tab. Select the components that you want to install.

Step 5 - Install language packs (optional)

By default, the installer program tries to match the language of the operating system when it runs for the first time. To install SSMS in a language of your choosing, choose the **Language packs** tab from the Visual Studio Installer, and then follow the prompts.

Change the installer language from the command line

Another way that you can change the default language is by running the installer from the command line. For example, you can force the installer to run in English by using the following command:

Windows Command Prompt

```
vs_ssms.exe --locale en-US
```

The installer remembers this setting when you run it again. The installer supports these language locales: `zh-cn`, `zh-tw`, `cs-cz`, `en-us`, `es-es`, `fr-fr`, `de-de`, `it-it`, `ja-jp`, `ko-kr`, `pl-pl`, `pt-br`, `ru-ru`, and `tr-tr`.

Step 6 - Select the installation location (optional)

You can reduce the installation footprint of SSMS on your system drive. For more information, see Select installation locations for SQL Server Management Studio 21 Preview.

You can select a different drive for **SQL Server Management Studio** or **Download cache** only when you first install SSMS. If you already installed it and want to change drives, you must uninstall SSMS and then reinstall it.

If SSMS or Visual Studio is already installed on your computer, you can't change the Shared components, tools, and SDKs path. It appears greyed out. This location is shared by all installations of SSMS and Visual Studio.

Step 7 - Sign in to your account (optional)

While you don't have to sign in, there are many advantages to doing so, such as synchronizing your settings across devices. See Sign in or switch user accounts in SQL Server Management Studio 21 Preview.

Support or troubleshooting

Sometimes, things can go wrong. If your SSMS installation fails, see Troubleshoot installation and upgrade issues for SQL Server Management Studio 21 Preview for step-by-step guidance.

You can reference the SSMS Developer Community site for other known issues (filter on **Tooling** under **Groups**) and to provide feedback to the product team.

Related content

- Update SQL Server Management Studio 21 Preview
- Modify SQL Server Management Studio 21 Preview components and language packs
- Uninstall or remove SQL Server Management Studio 21 Preview
- Create an offline installation of SQL Server Management Studio 21 Preview
- Use command-line parameters to install SQL Server Management Studio 21 Preview

---

Feedback

Was this page helpful? Yes No

Select installation locations for SQL Server Management Studio 21 Preview

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

You can reduce the installation footprint of SQL Server Management Studio (SSMS) on your system drive by changing the location of some of its files. Specifically, you can use a different location for the **download cache** and **shared components, tools, and SDKs**.

> **Note**
>
> There are some tools and SDKs that have different rules on where they can be installed. Such tools and SDKs are installed on your system drive even if you choose another location.

Get started

1. When you install SSMS, choose the **Installation locations** tab.
2. The default path is shown in **Product**. If you accept the default path, the core product and files that are specific to this version of SSMS are installed in that location.

   If your system drive is a solid-state drive (SSD), we recommend that you keep the core product on your system drive. When you use SSMS, you can read from and write to a lot of files, which increases the disk I/O activity. It's best to choose your fastest drive to handle the load.

   > **Important**
   >
   > You can select a different location only when you first install SSMS. If you've already installed it and want to change the location, you must uninstall SSMS and then reinstall it.

3. In the **Download cache** section, decide whether you want to keep the download cache after installation, and if so, where you want to store its files.

- Check or uncheck **Keep download cache after the installation**.

  If you decide not to keep the download cache, the download cache location is only temporarily used. This action doesn't affect or delete files from previous installations.

- Specify the folder path, including the drive, where you want to store the installation files and manifests from the download cache.

> **Important**
>
> You can select a different location only when you first install SSMS. If you've already installed it and want to change the location, you must uninstall SSMS and then reinstall it.

4. In the **Shared components, tools, and SDKs** section, select the folder where you want to store the files that are shared by side-by-side SSMS installations. SDKs and tools are also stored in this directory.

> **Important**
>
> If you've installed SSMS or Visual Studio on your computer before, you won't be able to change the **Shared components, tools, and SDKs** path and it will appear greyed out.

Support or troubleshooting

Sometimes, things can go wrong. If your SSMS installation fails, see Troubleshoot installation and upgrade issues for SQL Server Management Studio 21 Preview for step-by-step guidance.

You can reference the SSMS Developer Community site for other known issues (filter on **Tooling** under **Groups**) and to provide feedback to the product team.

Related content

- Install SQL Server Management Studio 21 Preview
- Update SQL Server Management Studio 21 Preview
- Modify SQL Server Management Studio 21 Preview components and language packs

- Uninstall or remove SQL Server Management Studio 21 Preview

Feedback

Was this page helpful? Yes No

Use command-line parameters to install SQL Server Management Studio 21 Preview

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

SQL Server Management Studio 21 Preview is a powerful tool for managing SQL Server instances, Azure SQL databases, and Azure SQL managed instances. This article provides detailed instructions on how to use command-line parameters to install SQL Server Management Studio 21 Preview, allowing for a more customized and automated installation process. Whether you're installing SQL Server Management Studio 21 Preview for the first time or updating an existing installation, these instructions provide the necessary steps to achieve your goals.

Command-line parameters

When you install SQL Server Management Studio (SSMS) programmatically or from a command prompt, you can use various command-line parameters to control or customize the installation to perform the following actions:

- Start the installation on the client with certain options and behaviors preselected.
- Automate the installation or update process.
- Create or maintain a network layout of the product files for installing or updating client machines.

The following command-line verbs and parameters can be used with the following executables or programs:

- The setup bootstrapper, which is the small (~1 MB) file (for example, vs_ssms.exe) that initiates the download process and the initial installation.
- The Visual Studio installer that might already be installed on the machine and is located in the folder `C:\Program Files (x86)\Microsoft Visual Studio\Installer\setup.exe`. You must initiate the installer programmatically from a *different* directory that the installer resides in. The installer is typically used for update or modify commands.

Not all commands or parameters work in each of these situations, and any special caveats or exceptions are documented. Furthermore, in some scenarios you might not have access to all of these executables described previously. For example, client machines might only have the installer available for programmatic execution if SSMS was installed via a layout.

To get the latest bootstrappers for SQL Server Management Studio 21 Preview that always install the latest version of the selected channel, download one of the files in the following table. Alternatively, to install a specific version or a specific channel of SQL Server Management Studio 21 Preview, see Release history for SQL Server Management Studio. This page has links to the fixed version bootstrappers for each servicing release.

Expand table

Channel	Version	Bootstrapper
Preview	21.0	SQL Server Management Studio 21 Preview

The Release channel will be available when SQL Server Management Studio 21 is generally available (GA).

> **Tip**
>
> If you previously downloaded a bootstrapper file and want to verify what version it installs, here's how. In Windows, open File Explorer, right-click the bootstrapper file, choose **Properties** and then choose the **Details** tab. The **Product version** field describes the **channel and version** that the bootstrapper installs. The version number should always be read as "latest servicing version of what is specified", and the channel is Release unless explicitly specified. A bootstrapper with a product version that says SQL Server Management Studio 21 Preview installs the latest version of SQL Server Management Studio 21 Preview on the Preview channel.

Install, update, modify, repair, uninstall, and export commands and command-line parameters

When the SSMS bootstrapper or the installer is invoked programmatically, to install the product or to maintain a layout:

- The first parameter is the command (the verb) that describes the operation to perform.
- The subsequent optional command line parameters, all prefixed by two dashes (`--`), further define how that operation is supposed to happen.

All SSMS command-line parameters are case-insensitive.

Syntax example: `vs_ssms.exe [command] <optional parameters>...`

Expand table

Command	Description
(blank)	The default command both installs the product, and it's used for all layout maintenance operations.
`modify`	Modifies an installed product.
`update`	Updates an installed product.
`updateall`	Updates all of the installed products in sequential order. Works with `--quiet` and `--passive` parameters.
`repair`	Repairs an installed product.
`uninstall`	Uninstalls an installed product.
`export`	Exports installation configuration into a `*.vsconfig` file.

> **Important**
>
> When specifying multiple distinct workloads, components, or languages, you must repeat the `--add` or `--remove` command-line switch for each item.

Expand table

Parameters	Description
`--installPath <dir>`	For the default install command, this parameter is **Optional** and describes where the instance is installed on the client machine. For other commands like update or modify, this parameter is **Required** and denotes the installation directory for the instance to act upon.
`--productId <id>`	**Optional**: The ID of the product instance that is modified, and used with `--channelId`. The `productId` is `Microsoft.VisualStudio.Product.SSMS`.

Parameters	Description
`--channelUri`	**Optional**: During an update command, you can pass in a new `--channelUri` to change the update settings location. Recommend to pair with `--installPath` parameter so that it's explicit which instance of SSMS you're configuring. See syntax examples of --channelUri
`--channelId <id>`	The ID of the channel, for example, `SSMS.21.SSMS.Preview`. `channelId` is required for modify operations, alongside either `--productId` or `--installPath`.
`--add <one or more workload or component IDs>`	**Optional**: During an install or modify command, this repeatable parameter specifies one or more workload or component IDs to add. The required components of the artifact are installed, but not the recommended or optional components. You can control other components globally using `--includeRecommended` and/or `--includeOptional` parameters. To include multiple workloads or components, repeat the `--add` command (for example, `--add Workload1 --add Workload2`). For finer-grained control, you can append `;includeRecommended` or `;includeOptional` to the ID (for example, `--add Workload1;includeRecommended` or `--add Workload2;includeRecommended;includeOptional`).
`--all`	**Optional**: During an install or modify command, this parameter causes all workloads and components for the product to be installed.
`--allWorkloads`	**Optional**: During an install or modify command, this parameter installs all workloads and components, but no recommended or optional components.
`--includeRecommended`	**Optional**: During an install or modify command, this parameter includes the recommended components for any workloads that are installed. It doesn't include the optional components. The workloads are specified either with `--allWorkloads` or `--add`. By using the `--includeRecommended` parameter, you enable your **Update Settings** dialog to "add recommended components for installed workloads on update". See Customize update settings to change this setting.
`--includeOptional`	**Optional**: During an install or modify command, this parameter includes the optional components for any workloads that are installed. It doesn't include the recommended components. The workloads are specified either with `--allWorkloads` or `--add`.
`--addProductLang <language-locale>`	**Optional**: During an install or modify command, this repeatable parameter specifies the UI language packs that should be installed with the product. If not present, the installation uses the language pack that corresponds to the machine locale. For more information, see the [List of language locales]

Parameters	Description
`--remove <one or more workload or component IDs>`	**Optional**: During a modify command, this repeatable parameter specifies one or more workload or component IDs to remove. It complements and behaves similarly to the `--add` parameter.
`--addProductLang <language-locale>`	**Optional**: During an install or modify command, this repeatable parameter specifies the UI language packs that should be installed with the product. If not present, the installation uses the language pack that corresponds to the machine locale. For more information, see the List of language locales section on this page.
`--removeProductLang <language-locale>`	**Optional**: During an install or modify command, this repeatable parameter determines the UI language packs that should be removed from the product. It complements and behaves similarly to the `--addProductLang` parameter.
`--in <path>`	**Optional**: The URI or path to a response file, which can contain configuration settings.
`--quiet`	**Optional**: Used with any command, this parameter prevents any user interface from being displayed while the command is being executed. Not available to use programmatically by standard users regardless of how the AllowStandardUserControl policy is set.
`--passive, -p`	**Optional**: This parameter causes the user interface to be displayed in a non-interactive manner. This parameter is mutually exclusive from (and in fact overrides) the `--quiet` parameter. Also not available to use programmatically by standard users regardless of how the AllowStandardUserControl policy is set.
`--norestart`	**Optional**: This parameter must be paired with either the `--passive` or `--quiet` parameters. During an install, update, or modify command, adding the `--norestart` parameter delays any necessary reboot.
`--force`	**Optional**: This parameter forces SSMS to close even if any SSMS process is in use. Forcing SSMS to close might cause loss of work, so use it with caution.
`--installWhileDownloading`	**Optional**: During an install, update, or modify command, this parameter allows SSMS to both download and install the product in parallel. It's the default experience.
`--downloadThenInstall`	**Optional**: During an install, update, or modify command, this parameter forces SSMS to download all files before installing them. It's mutually exclusive from the `--installWhileDownloading` parameter.
`--nickname <name>`	**Optional**: During an install command, this parameter defines the nickname to assign to an installed product. The nickname can't be

Parameters	Description
	longer than 10 characters.
`--removeOos true`	**Optional**: During an install, update, or modify command, this parameter, followed by `true` or `false`, tells the Visual Studio Installer whether to remove all installed components transitioned to an out-of-support state. This behavior is applicable for a single event. If you want to make this behavior persistent, apply this parameter to the `modifySettings` command, described later, or configure the `removeOOS` global policy. Useful for helping to keep the machine secure.
`--config <path to *.vsconfig file>`	**Optional**: During an install or modify operation, you can pass in a configuration file using the `--config` parameter to specify the workloads, components, or extensions to add based on a previously saved installation configuration file. This operation is additive only; it doesn't remove anything not specified in the file. Also, items specified in the config file that don't apply to the product aren't added. Highly recommend you specify the fully qualified path to the config file. During an export operation, this parameter determines the location to save the installation configuration file.
`--allowUnsignedExtensions`	**Optional**: During an install or modify operation run in a `--passive` or `--quiet` context using a `--config` file, if extensions are specified in the config file, then this parameter is necessary in order to load extensions without a digital signature.
`--installerOnly`	**Optional**: During an install or update operation, this parameter tells the Visual Studio Installer to install just itself (the installer) and not the SSMS product. Its functionality is equivalent to the `--update` parameter, but more intuitive. Meant to be used when prepping and preconfiguring client machines.
`--help, --?, -h, -?`	Displays an offline version of this page.

Layout command and command-line parameters

All layout management operations, such as creating or updating a layout, are run using the bootstrapper exe and assume that the command is the default Install (blank). All layout management operations should start with the required `--layout` initial parameter. The following table describes the other parameters you can use to create or update a layout using the command line.

Expand table

Layout parameters	Description
`--layout <dir>`	Specifies a directory to create or update an offline install cache.
`--lang <one or more language-locales>`	**Optional**: Used with `--layout` to prepare an offline install cache with resource packages with the specified one or more specified languages. For more information, see the List of language locales section on this page.
`--add <one or more workload or component IDs>`	**Optional**: One or more workload or component IDs to add. The required components of the artifact are installed, but not the recommended or optional components. You can control more components globally using `--includeRecommended` and/or `--includeOptional`. For finer-grained control, you can append `;includeRecommended` or `;includeOptional` to the ID (for example, `--add Workload1;includeRecommended` or `--add Workload2;includeOptional`). **Note**: If `--add` is used, only the specified workloads and components and their dependencies are downloaded. If `--add` isn't specified, all workloads and components are downloaded to the layout.
`--includeRecommended`	**Optional**: Includes the recommended components for any workloads that are installed, but not the optional components. The workloads are specified either with `--allWorkloads` or `--add`.
`--includeOptional`	**Optional**: Includes the recommended *and* optional components for any workloads being included in the layout. The workloads are specified with `--add`.
`--wait`	**Optional**: The process waits until the install is completed before returning an exit code. `wait` is useful when automating installations where one needs to wait for the install to finish to handle the return code from that install. The `--wait` parameter can only be passed into the bootstrapper; the installer (setup.exe) doesn't support it. It's useful when updating layouts.
`--config <path to *.vsconfig file>`	**Optional**: If present, SSMS uses the contents of the configuration file to configure your layout. Make sure you specify the fully qualified path to the config file.
`--noWeb`	**Optional**: If present, SSMS setup uses the files in your layout directory to install SSMS, and it doesn't download any packages from the web. If a user tries to install components that aren't in the layout, setup fails. Don't use this parameter if you're trying to deploy a layout hosted on an internal intranet website. **Important**: The `--noWeb` parameter doesn't stop the Visual Studio Installer on an internet-connected client machine from checking for updates if the client is configured to look at Microsoft hosted servers for

Layout parameters	Description
	updates. In this case, `--noWeb` simply prevents the client from downloading the product packages.
`--verify`	**Optional**: Verify the contents of a layout. Any corrupt or missing files are listed.
`--fix`	**Optional**: Verify the contents of a layout. If any files are corrupt or missing, they're redownloaded. Internet access is required to fix a layout.
`--clean <one or more paths to catalogs>`	**Optional**: Removes old versions of components from a layout that is updated to a newer version.

Expand table

Advanced layout parameters	Description
`--channelId <id>`	**Optional**: The ID of the channel for the instance to be installed, represented like `SSMS.21.SSMS.Preview`. `channelId` is required for the install command, and ignored for other commands if `--installPath` is specified. For more information, see syntax examples of `--channelId`.
`--channelUri <uri>`	**Optional**: The URI of the channel manifest. This value governs the source location of updates and the initial value is configured in the layout's `response.json` file. If updates aren't wanted, `--channelUri` can point to a nonexistent file (for example, `--channelUri C:\doesntExist.chman`). This parameter can be used for the install command; other commands ignore it.
`--installChannelUri <uri>`	**Optional**: The URI of the channel manifest to use for the installation. The URI specified by `--channelUri` (which must be specified when `--installChannelUri` is specified) is used to detect updates. This parameter can be used for the install command; other commands ignore it.
`--installCatalogUri <uri>`	**Optional**: The URI of the catalog manifest to use for the installation. If specified, the channel manager attempts to download the catalog manifest from this URI before using the URI in the install channel manifest. This parameter is used to support offline install, where the layout cache is created with the product catalog already downloaded. This parameter can be used for the install command; other commands ignore it.
`--productId <id>`	**Optional**: The ID of the product for the instance that is installed. This parameter is prepopulated in normal installation conditions. The `productId` is `Microsoft.VisualStudio.Product.SSMS`.
`--keepLayoutVersion`	**Optional**: Apply changes to the layout without updating the product version that's included in the layout.

Advanced layout parameters	Description
`--locale <language-locale>`	**Optional**: Change the display language of the user interface for the installer itself. The setting is persisted. For more information, see the List of language locales section on this page.
`--cache`	**Optional**: If present, packages will be kept after being installed for subsequent repairs. This overrides the global policy setting to be used for subsequent installs, repairs, or modifications. The default policy is to cache packages. This is ignored for the uninstall command.
`--nocache`	**Optional**: If present, packages will be deleted after being installed or repaired. They'll be downloaded again only if needed and deleted again after use. This overrides the global policy setting to be used for subsequent installs, repairs, or modifications. The default policy is to cache packages. This is ignored for the uninstall command.
`--noUpdateInstaller`	**Optional**: If present, prevents the installer from updating itself when `quiet` is specified. The installer fails the command and returns a non-zero exit code if `--noUpdateInstaller` is specified with `quiet` when an installer update is required.
`--path <name>=<path>`	**Optional**: Used to specify custom install paths for the installation. Supported path names are shared, cache, and install.
`--path cache=<path>`	**Optional**: Uses the location you specify to download installation files. This location can only be set the first time that SSMS is installed. Example: `--path cache="C:\SSMS\cache"`
`--path shared=<path>`	**Optional**: Contains shared files for side-by-side SSMS and Visual Studio installations. Some tools and SDKs install to a location on this drive, while some others might override this setting and install to another drive. Example: `--path shared="C:\VS\shared"` **Important**: This can be set only once and on the first time that either SSMS or Visual Studio is installed.
`--path install=<path>`	**Optional**: Equivalent to `--installPath`. Specifically, `--installPath "C:\SSMS"` and `--path install="C:\SSMS"` are equivalent. Only one of these commands can be used at a time.

ModifySettings command and command-line parameters

You can modify the update settings and programmatically configure the source location of updates for a given instance of SSMS by using either the installer or the bootstrapper

on the client machine, and passing in the modifySettings command and the desired update channel.

Expand table

Command	Description
`modifySettings`	Verb used to modify the update settings of a particular instance of SSMS.

Expand table

modifySettings parameters	Description
`--installPath <dir>`	**Recommended** to use to specify which instance of SSMS to act upon.
`--newChannelUri`	**Required**: The URI of the channel manifest. This value specifies where the next source location of updates will be. If updates aren't wanted, `--channelUri` can point to a nonexistent file (for example, `--channelUri C:\doesNotExist.chman`).
`--channelUri`	The URI of the old channel manifest. Can be used if the `--installPath` isn't known. Must be used with `--productId` to identify the right instance to act upon.
`--productId <id>`	Must be used if `--channelUri` is specified and is used to identify the right instance to act upon. The `productId` is `Microsoft.VisualStudio.Product.SSMS`.
`--includeRecommended`	**Optional**: During a modifySettings command, this parameter (which must have the word true or false immediately after it) tells the SSMS installer to disable the "add recommended components for installed workloads on update" functionality in the Update Setting dialog.
`--quiet`	**Optional**: This parameter prevents any user interface from being displayed while the command is being executed. Not available to use programmatically by standard users.
`--removeOos true`	**Optional**: During a modifySettings command, this parameter (which must have the word true or false immediately after it) tells the SSMS installer to *persistently* remove (or not remove) all installed components that have transitioned to an out-of-support state. Useful for helping to keep the machine secure.

Syntax examples:

Windows Command Prompt

```
C:\>"C:\Program Files (x86)\Microsoft Visual Studio\Installer\setup.exe"
modifySettings --installPath "C:\Program Files\Microsoft SQL Server
Management Studio 21\Preview" --newChannelUri
https://aka.ms/ssms/21/preview/channel --removeOos true
```

Windows Command Prompt

```
 C:\>"C:\Program Files\Microsoft SQL Server Management Studio
21\Preview\vs_ssms.exe" modifySettings --channelUri
https://aka.ms/ssms/21/preview/channel --productId
Microsoft.SSMS.Product.Community --newChannelUri
\\layoutserver\share\path\channelmanifest.json --removeOos true --quiet
```

Remove channel command and command-line parameters

Channels that are available to update from, are cached on the client, and over time they can generate clutter. You can manually remove update channels by bringing up the Visual Studio Installer, switching to the **Available** tab, and selecting on the **X** in the top right corner of the product card. You can programmatically remove channels (for example, older layout locations) using the `removeChannel` command. You can run `vswhere` programmatically on the client machine to determine which channels are cached on the client machine.

Expand table

Command	Description
`removeChannel`	Command to remove a channel from the client machine.

Expand table

removeChannel parameters	Description
`--channelUri`	**Required** The URI of the old channel manifest.
`--quiet`	**Optional** This parameter prevents any user interface from being displayed while the command is being executed.
`--passive`	**Optional** This parameter overrides the `--quiet` parameter. It causes the UI to appear in a non-interactive manner.

Syntax example:

Windows Command Prompt

```
"C:\Program Files (x86)\Microsoft Visual Studio\Installer\setup.exe"
removeChannel --channelUri
"\\\\server\\share\\layoutdirectory\\ChannelManifest.json"
```

List of language locales

The following table lists the language locales that can be used with the `--addProductLang` and `--removeProductLang` parameters.

Expand table

Language-locale	Language
`cs-cz`	Czech
`de-de`	German
`en-us`	English
`es-es`	Spanish
`fr-fr`	French
`it-it`	Italian
`ja-jp`	Japanese
`ko-kr`	Korean
`pl-pl`	Polish
`pt-br`	Portuguese - Brazil
`ru-ru`	Russian
`tr-tr`	Turkish
`zh-cn`	Chinese - Simplified
`zh-tw`	Chinese - Traditional

Error codes

Depending on the result of the operation, the `%ERRORLEVEL%` environment variable is set to one of the following values:

Expand table

Value	Result
0	Operation completed successfully
740	Elevation required
1001	Visual Studio installer process is running
1003	Visual Studio is in use
1602	Operation was canceled
1618	Another installation running
1641	Operation completed successfully, and reboot was initiated
3010	Operation completed successfully, but install requires reboot before it can be used
5003	Bootstrapper failed to download installer
5004	Operation was canceled
5005	Bootstrapper command-line parse error
5007	Operation was blocked - the computer doesn't meet the requirements
8001	Arm machine check failure
8002	Background download precheck failure
8003	Out of support selectable failure
8004	Target directory failure
8005	Verifying source payloads failure
8006	Visual Studio processes running
8010	Operating System not supported. Refer to the system requirements.
-1073720687	Connectivity failure
-1073741510	Microsoft Visual Studio Installer was terminated (by the user or external process)
Other (for example: -1, 1, 1603)	Failure condition occurred - check the logs for more information

Each operation generates several log files in the `%TEMP%` directory that indicate the progress of the installation. Sort the folder by date and look for files that begin with `ssms_bootstrapper`, `ssms_client`, and `ssms_setup` for the bootstrapper, the installer app, and the setup engine, respectively.

Support or troubleshooting

Sometimes, things can go wrong. If your SSMS installation fails, see Troubleshoot installation and upgrade issues for SQL Server Management Studio 21 Preview for step-by-step guidance.

You can reference the SSMS Developer Community site for other known issues (filter on **Tooling** under **Groups**) and to provide feedback to the product team.

Related content

- Create an offline installation of SQL Server Management Studio 21 Preview

---

Feedback

Was this page helpful? Yes No

Create an offline installation of SQL Server Management Studio 21 Preview

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

SQL Server Management Studio 21 Preview is designed to work well in various computer configurations. In this article, you learn how to create an offline installation package of files for installation on the *local machine*.

Use the "Download all, then install" feature

Sometimes online access is problematic. For example, you might have an unreliable internet connection or your internet connection might have low bandwidth. For situations like these, you have other methods available for acquiring SQL Server Management Studio (SSMS). You can use the **Download all, then install** feature from the Visual Studio Installer to download an installation package on the local machine *before* you install it locally, or you can use the command line to create a local installation package to install locally later.

After you download the bootstrapper, run it to install SSMS. It first installs and then launches the latest version of the Visual Studio Installer, which you can use to customize and configure your installation, download installation packages, and install the product.

To complete downloading the product before installation starts, select the **Download all, then install** option in the dropdown list at the bottom of the default **Workloads** tab of the Visual Studio Installer. The purpose of this feature is to download the SSMS packages in advance on the computer where SSMS is eventually installed. By downloading the packages locally first, you can then safely disconnect from the internet before you install SSMS.

Use the command line to create a local layout

Step 1 - Download the SSMS bootstrapper

Download the correct bootstrapper for the version of SSMS you want, and copy it into the directory you want to use as the source location for your local layout. The

bootstrapper is the executable you use to create, update, or modify your local layout. You must have an internet connection to complete this step.

Step 2 - Create a local layout

Open a command prompt with administrator privileges, navigate to the directory where you downloaded the bootstrapper, and use the bootstrapper's parameters to create your local layout. You must have an internet connection to complete this step.

You can install a language other than English by changing `en-US` to a locale from the list of language locales, and you can use the list of component IDs to further customize your local layout.

To create a complete local layout for SQL Server Management Studio, run:

Windows Command Prompt

```
vs_ssms.exe --layout c:\localSSMSlayout --add Microsoft.Component.HelpViewer
```

After the bootstrapper has finished downloading the layout files, the local layout folder can be moved to any other machine or environment you want to install SQL Server Management Studio on without requiring an internet connection.

> **Note**
>
> Make sure that your full installation path is less than 80 characters and that your machine has ample storage. SQL Server Management Studio requires a minimum of 4-GB of disk space. For more information, see System requirements for SQL Server Management Studio 21 Preview.

Step 3 - Install SSMS from the local layout

When you install SSMS from a local layout, the Visual Studio Installer uses the local versions of the files. If you select components during installation that aren't in the layout, then the Visual Studio Installer attempts to download them from the internet. To make sure you install only the files you previously downloaded, use the same command-line options you used to create the local layout. To make sure your installer doesn't try to access the internet when it's installing the product, use the `--noweb` switch.

For example, if you created a local installation layout using the command from step 2, then use the following command to run the installation and prevent the client machine

from accessing the internet:

Windows Command Prompt

```
c:\localSSMSlayout\vs_ssms.exe --noWeb --add Microsoft.Component.HelpViewer
```

Support or troubleshooting

Sometimes, things can go wrong. If your SSMS installation fails, see Troubleshoot installation and upgrade issues for SQL Server Management Studio 21 Preview for step-by-step guidance.

You can reference the SSMS Developer Community site for other known issues (filter on **Tooling** under **Groups**) and to provide feedback to the product team.

Related content

- Use command-line parameters to install SQL Server Management Studio 21 Preview
- Install SQL Server Management Studio 21 Preview
- Modify SQL Server Management Studio 21 Preview components and language packs

---

Feedback

Was this page helpful? Yes No

Install local help documentation for SQL Server Management Studio 21 Preview

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

By using the Microsoft Help Viewer, you can add, remove, update, and move the Help content that is installed on your computer for use with SQL Server Management Studio (SSMS).

To manage content on your local computer, you must sign in with an account that has administrator permissions.

Install the Help Viewer

In order to manage content on your local computer, you must have the Help Viewer installed. The Help Viewer is a component available during installation.

To modify your SSMS installation to install the Help Viewer, see Modify SQL Server Management Studio 21 Preview components and language packs.

Change installation source

By default, Help Viewer installs content by using a Microsoft online service as the source. You generally shouldn't change your content source unless you work in an enterprise environment for which a system administrator has already installed content in another location.

Change the content installation source

To open the Help Viewer, open SSMS and then select **Help** > **Add and Remove Help Content**.

1. On the **Manage Content** tab, choose the **Disk** option button.

> **Note**
>
> The **Disk** option isn't available if your administrator has prevented you from modifying the content installation source. For more information, see the Help

Viewer administrator guide.

2. Perform one of the following steps:

   - Enter the path of an `.msha` file or the URL of a service endpoint.
   - Choose the Browse (...) button to navigate to an `.msha` file.
   - In the list, choose the entry that was used most recently.

Download and install content locally

If you download and install content on your local computer, you can view articles when you don't have an internet connection.

> **ⓘ Important**
>
> To install content, you must log on with an account that has administrative permissions.

> **ⓘ Note**
>
> If SSMS is set to a language other than English, you can install English content, localized content, or both. However, no content appears if you install only the English version and the **Include English content in all navigation tabs and F1 requests** check box in the **Viewer Options** dialog box is cleared.

Download and install content

1. Choose the **Manage Content** tab.
2. In the content list, choose the **Add** link next to the book or books that you want to download and install.

   The book is added to the **Pending changes** list, and the estimated size of the book or books that you specified appear below that list. Because some books share articles, the total download size of multiple books might be smaller than the result of adding together the sizes of every book that you specified.
3. Choose the **Update** button.

The book or books that you specified are installed along with any updates for books that you already have on your computer. Installation times vary, but you can view the progress in the status bar.

Remove local content

You can save disk space by removing unwanted content from your computer.

> **Important**
>
> You must have administrative permissions to remove content.

> **Note**
>
> No content appears if SSMS is set to a language other than English, you remove localized content, and the **Include English content in all navigation tab and F1 requests** check box in the **Viewer Options** dialog box is cleared.

Remove content

1. Choose the **Manage Content** tab.
2. In the content list, choose the **Remove** link next to the book or books that you want to remove.

   The book is added to the **Pending changes** list.
3. Choose the **Update** button.

   The book or books that you specified are removed from your computer.

Update local content

The status bar indicates when updates to your installed content are available.

> **Important**
>
> If you want the **Help Viewer** to automatically check for online updates, you must open the **Viewer Options** dialog box and then select the **Go online to check for content updates** check box.

Update local content

- In the lower-right corner of the status bar, choose the **Click here to download now** link.

Update times can vary, but you can view the update progress in the status bar.

Move local content

You can save disk space by moving installed content from your local computer to a network share or to another partition on your local computer.

> **Important**
>
> To move content, you must log on with an account that has administrative permissions.

Move content

1. On the **Manage Content** tab, choose the **Move** button under **Local Store Path**.

   The **Move Content** dialog box opens.

2. In the **To** text box, enter a different location for the content, and then choose the **OK** button.

3. Choose the **Close** button when the content has moved.

Support or troubleshooting

Sometimes, things can go wrong. If your SSMS installation fails, see Troubleshoot installation and upgrade issues for SQL Server Management Studio 21 Preview for step-by-step guidance.

You can reference the SSMS Developer Community site for other known issues (filter on **Tooling** under **Groups**) and to provide feedback to the product team.

Related content

- Microsoft Help Viewer
- Override Help Viewer defaults

Feedback

Was this page helpful? Yes No

Install SQL Server Management Studio versions side-by-side

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

You can install SQL Server Management Studio 21 Preview on a computer that has an earlier major version of SQL Server Management Studio (SSMS) already installed.

Before you install versions side-by-side, review the following conditions:

- Suppose that you use SSMS 21 to open a solution that was created in SSMS 20 or an earlier version. You can later open and modify the solution again in the earlier version unless you implement features that are specific to SSMS 21.
- Suppose that you use SSMS 21 to open a solution that was created in SSMS 20 or an earlier version. You might need to modify your projects and files to be compatible with SSMS 21.
- If you uninstall a version of SSMS on a computer that has more than one version installed, the file associations for SSMS are removed for all versions.
- SSMS doesn't automatically upgrade extensions, because not all extensions are compatible. You must reinstall any extensions.

Install different channels of the same major SSMS version side-by-side

Each installation of SQL Server Management Studio 21 must have a unique combination of major version and channel. For example, you can install SQL Server Management Studio 21 Preview alongside the generally available (GA) SQL Server Management Studio 21 Release, when it's available.

With SQL Server Management Studio 21 Preview and later versions, when you upgrade from one minor version of SSMS to the next, by default, the Visual Studio Installer updates your current installation to the latest version in that channel. For example, if Microsoft releases version 21.1 to the preview channel, the installer tries to replace your current installation of SSMS with this latest version on the same channel.

Manual installation

You can manually use another bootstrapper to install a new instance of SSMS, or you can select one of the options from the Visual Studio Installer's **Available** tab.

1. To use another bootstrapper, you can download and run the bootstrapper file for SSMS, from either the installation article or the releases article, for the version that you want to install side-by-side with your existing version of SSMS.

2. Using the installer's **Available** tab presumes that you already have another channel of SSMS installed. First find the **Visual Studio Installer** on your computer and launch it. After it updates itself, select the **Available** tab and install one of the offered products.

Then follow the steps to customize your installation. For more information, see Install SSMS.

Support or troubleshooting

Sometimes, things can go wrong. If your SSMS installation fails, see Troubleshoot installation and upgrade issues for SQL Server Management Studio 21 Preview for step-by-step guidance.

You can reference the SSMS Developer Community site for other known issues (filter on **Tooling** under **Groups**) and to provide feedback to the product team.

Related content

- Install SQL Server Management Studio 21 Preview
- Update SQL Server Management Studio 21 Preview
- Modify SQL Server Management Studio 21 Preview components and language packs
- Uninstall or remove SQL Server Management Studio 21 Preview

---

Feedback

Was this page helpful? Yes No

Update SQL Server Management Studio 21 Preview

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

SQL Server Management Studio 21 Preview must be installed on the machine in order to update it.

Before you update

By default, in order to install, update, or modify SSMS, the account performing the update must have administrative permissions on the machine, and also must have permissions to the source where the product updates are located (either the internet or an internal location). If you're logged in as a typical user and try to perform one of these commands, then you get a User Account Control notice prompting you for admin credentials.

SQL Server Management Studio (SSMS) must be closed before performing an update. We highly recommend saving your work before performing an update.

We encourage you to update to the most recent release of SQL Server Management Studio (SSMS) so that you always get the latest features, security fixes, and improvements.

There are many different ways to update an installation of SSMS:

- Update using the Visual Studio Installer
- Within SSMS, check for updates or use the notification hub
- Configure SSMS to automatically update upon close
- Update by running a specific version of the bootstrapper
- Update SSMS programmatically.

Here's how to update SQL Server Management Studio (SSMS) using these various methods.

Use the Visual Studio Installer

1. Find the **Visual Studio Installer** on your computer.

In the Windows Start menu, search for "installer", and then select **Visual Studio Installer** from the results.

If you're prompted to update the Visual Studio Installer before continuing, do so by following the prompts.

> **Note**
>
> SSMS setup features and bug fixes are typically implemented in the installer. The installer ships the most current release for the latest supported SSMS version. This means that in general (unless otherwise specified), all new setup functionality is automatically available and works seamlessly in older versions of the product.

1. In the Visual Studio Installer, look for the installation of SSMS that you want to update.

   For example, if you previously installed SQL Server Management Studio 21 Preview and there's an update for it, then an **Update available** message appears in the Visual Studio Installer.

2. Choose **Update** to install the update.

3. After the update is complete, the Visual Studio Installer might prompt you to restart your computer. If so, do so, and then start SSMS as you typically would.

   If you aren't asked to restart your computer, choose **Launch** to start SSMS from the Visual Studio Installer.

Use the message box in the application

1. When you open SSMS, the application checks to see if an update is available. In certain situations, a **SQL Server Management Studio update** message will briefly appear. If you want to update immediately, select **View details**. If you want to defer the update until when you close SSMS, select **Update on Close**.

2. If you select **View details**, then in the subsequent **Update available** dialog, select **Update** to update now.

Use the Notifications hub in the application

1. Choose the notification icon from the lower-right corner of SSMS to open the **Notifications hub**.

2. In the **Notifications hub**, select the update that you want to install. If you want to update immediately, select **View details**. If you want to defer the update until when you close SSMS, select **Update on Close**.

3. If you select **View details**, then in the subsequent **Update available** dialog, select **Update**.

> **Note**
>
> The **Update on Close** command that is found in both the update notification message box or in the **Notification** hub isn't a permanent setting; it applies only to the current update. It's a way to defer the update to when you voluntarily close SSMS. To learn how to configure this setting permanently, see the Customize update settings section.

Manually check for updates

1. You can check to see if an update is available by choosing **Help** from the menu bar, and then choosing **Check for Updates**. You can also use the search box by selecting **Ctrl+Q**, typing **check for updates**, and then select the search result that matches. In the subsequent **Update available** dialog box, choose **Update**.

Run a specific bootstrapper to update the product to a specific version

You can update your instance of SQL Server Management Studio to any specific version that has been released, as long as it's a higher version than what is currently installed. To update your instance of SQL Server Management Studio via this method, see the release history, download the bootstrapper that corresponds to the desired update version into your product installation directory, and then double-click on the file to initiate the update.

Customize update settings

There are several different settings that can be customized to control the update behavior, such as how and when the product bits are downloaded and installed, or where the update source is located.

Always Update on Close and other download behaviors

1. On the menu bar, select **Tools** > **Options** > **Environment** > **Product Updates**. Observe the configuration options that are available to set in this dialog.

- You can select the **Automatically download updates** setting, which allows updates to download while your machine is idle.

- You can configure SSMS to automatically apply the latest available update upon close. This setting is configurable on a per-installation basis of SSMS. For example, you can set your Preview installation to apply updates when SSMS closes, but your Release installation can continue to be updated on demand. After you configure SSMS to **Always update on close**, updates begin once SSMS and all related processes are closed. If any extensions have been scheduled for installation or update, the SSMS update doesn't start until the next time SSMS closes.

- There are also two installation modes to choose from: **Install while downloading** and **Download all, then install**.

Configure source location of updates

With SQL Server Management Studio 21 Preview, you can now configure the location from where your clients get their updates. These update source locations are called *channels*, and you can find more information about channel purpose and availability in the Channels for SQL Server Management Studio 21 Preview documentation. Microsoft makes both the Release and the Preview channels available to everyone.

There are two ways to bring up the Update Settings dialog, which allows you to manually change the channel from which your SSMS installation should get its updates.

1. Open the Visual Studio Installer, select the installation you want to configure, select **More**, and then select the **Update settings** menu option. Refer to previous instructions for how to find the Visual Studio Installer.

2. An alternative way to invoke the Update Settings dialog is to open SSMS, bring up the Update available dialog (either **View details** on an update notification or **Check for updates** on the Help menu) and select the **Change update settings** link.

By selecting the correct value in the **Update channel** dropdown list, you can control the source location of future updates for this installation of SSMS. Also keep in mind the following items:

- The Preview and Release channels are available for SQL Server Management Studio 21 and higher.

- You can choose to update your installation of SSMS immediately after you configure the **Update channel** location, or you can defer the actual product update until a later time. The act of configuring the update channel and the act of updating the product are two independent events. For information on how to programmatically control the update channel, refer to Use the modifySettings command.

- You can only change the update channel if the version of the product that's available in the channel is **greater** than the version you installed. For example, you can always transition from the Release channel to the Preview channel, but you can't transition from the Preview channel to the Release channel until the latest release on the Release channel surpasses the version of Preview that you have installed.

- When you update to a new channel, you install the most recent release on that channel. If you want to install a particular version of the product on that channel, then follow the Run a specific bootstrapper instructions described previously.

- All Microsoft channels are hosted on Microsoft servers and require access to the internet.

- Each installation of SSMS has the ability to independently configure its source for updates. If you have two SQL Server Management Studio (SSMS) installations, each can update from a different channel.

Use the latest installer

The most current release of the installer in the latest version of SSMS typically has the most robust and reliable feature set. This functionality often seamlessly works on lower versioned products. In order to manually acquire the latest version of the installer onto a client machine, you can do one of the following options:

1. Download any recent bootstrapper for the latest version of SSMS and then either:

    - Run the bootstrapper to initiate the installation. The latest installer is installed onto the machine first. Once the **Choose Workloads** tab appears, cancel the installation.

    - In a command window, use the bootstrapper to update the installer:

        Windows Command Prompt

        ```
        vs_ssms.exe --update --quiet
        ```

Support or troubleshooting

Sometimes, things can go wrong. If your SSMS installation fails, see Troubleshoot installation and upgrade issues for SQL Server Management Studio 21 Preview for step-by-step guidance.

You can reference the SSMS Developer Community site for other known issues (filter on **Tooling** under **Groups**) and to provide feedback to the product team.

Related content

- Install SQL Server Management Studio versions side-by-side
- Modify SQL Server Management Studio 21 Preview components and language packs
- Uninstall or remove SQL Server Management Studio 21 Preview

---

Feedback

Was this page helpful? Yes No

Modify SQL Server Management Studio 21 Preview components and language packs

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

It's easy to modify SQL Server Management Studio 21 Preview so that it includes only what you want. To do so, open the Visual Studio Installer, and then add or remove workloads, components, and language packs. This article walks you through the process.

Prerequisites

- By default, to install, update, or modify SQL Server Management Studio (SSMS), the account performing the update must have administrative permissions on the machine. It must also have permissions to the source where the product updates are located (either the internet or an internal location). If you're logged in as a typical user and try to perform one of these commands, you get a User Account Control notice prompting you for admin credentials. An admin can delegate control of modifying SSMS to standard users by configuring the AllowStandardUserControl policy. For more information, see User Permissions and Visual Studio.
- The following procedures assume you have an internet connection. For information about how to modify a previously created offline installation of SSMS, see:
  - Update a network-based installation of SSMS
  - Control updates to network-based SSMS deployments

Open the Visual Studio Installer to modify your installation

To modify your SSMS installation, you need to open the Visual Studio Installer, and then select an SSMS installation to modify.

1. There are many ways to open the Visual Studio Installer:

- On the Windows Start menu, you can search for "installer" and then select **Visual Studio Installer** from the results.
- Run the **Visual Studio Installer** executable file, which is located at `C:\Program Files (x86)\Microsoft Visual Studio\Installer\setup.exe`.

You might be prompted to update the Visual Studio Installer before continuing. If you are, follow the prompts.

In the Visual Studio Installer, look for the installation of SSMS that you want to modify and then select **Modify**.

Change individual components

SSMS contains the components you need for managing SQL Server and Azure SQL databases.

1. In the Visual Studio Installer, on the **Workloads** tab, select the workloads that you want.
2. To add individual components, on the **Individual components** tab, select the individual components that you want.
3. You can select the default **Install while downloading** or **Download all, then install**.
4. Select **Modify**.
5. After the modified workloads or components are installed, select **Launch** in the Visual Studio Installer to open SQL Server Management Studio.

You can also use a configuration file to modify an existing installation to add or remove components. For more information, see Import or export installation configurations.

> **Tip**
>
> For information about the SQL Server Data Tools (SSDT) component, see Install SQL Server Data Tools (SSDT) for Visual Studio.

Modify language packs

The Visual Studio Installer selects a default language pack for SSMS that matches the language of your operating system. You can change the default language whenever you

want.

To change the default language:

1. Select the **Language packs** tab in the Visual Studio Installer.
2. Select the language you prefer.
3. Follow the prompts.

Support or troubleshooting

Sometimes, things can go wrong. If your SSMS installation fails, see Troubleshoot installation and upgrade issues for SQL Server Management Studio 21 Preview for step-by-step guidance.

You can reference the SSMS Developer Community site for other known issues (filter on **Tooling** under **Groups**) and to provide feedback to the product team.

Related content

- Update SQL Server Management Studio 21 Preview
- Uninstall or remove SQL Server Management Studio 21 Preview
- Repair installation for SQL Server Management Studio 21 Preview

---

Feedback

Was this page helpful? Yes No

Repair installation for SQL Server Management Studio 21 Preview

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

Your SQL Server Management Studio (SSMS) installation could become damaged or corrupted. A repair is useful for fixing a wide variety of install-time issues, including update issues.

Repairing SSMS resets its environment. Repairing removes local customizations such as extensions installed without elevation, user settings, and profiles. It restores your synchronized settings such as themes, colors, and key bindings.

When to use repair

Use repair if you're having issues with:

- Installation payload, which can happen when writing a file to disk is unsuccessful and the file gets corrupted. Repair can reacquire needed files.
- Client-side download, assuming that you fixed any internet connection or proxy issues.
- Updating SQL Server Management Studio. Repair fixes many common update issues.

> **Tip**
>
> An unstable internet connection or an issue in a Windows service, like Windows Installer, can cause install issues. In those scenarios, repair might also be affected. To check for underlying issues, review the error report generated by the Visual Studio Installer.

Repairing SSMS resets user settings and reinstalls your existing assemblies. If you experience a product issue and repair doesn't fix it, see How to report a problem with Visual Studio Installer.

How to repair

To repair your installation, follow these steps.

1. Find the **Visual Studio Installer** on your computer.

2. In the Windows Start menu, you can search for installer, and then select **Visual Studio Installer** from the results.

   > ⓘ **Note**
   >
   > You can also find the Visual Studio Installer in the following location:
   >
   > `C:\Program Files (x86)\Microsoft Visual Studio\Installer\setup.exe`

   You might be prompted to update the Visual Studio Installer before you continue. If so, follow the prompts.

3. In the Visual Studio Installer, look for the installation of SQL Server Management Studio (SSMS) that you want to repair. Then choose **Repair** from the More dropdown list menu.

   The **Repair** option appears only for installed instances of SSMS. If you don't see the **Repair** option, check whether you selected **More** in a version that appears in the Visual Studio Installer as **Available** rather than **Installed**.

Support or troubleshooting

Sometimes, things can go wrong. If your SSMS installation fails, see Troubleshoot installation and upgrade issues for SQL Server Management Studio 21 Preview for step-by-step guidance.

You can reference the SSMS Developer Community site for other known issues (filter on **Tooling** under **Groups**) and to provide feedback to the product team.

Related content

- Install SQL Server Management Studio 21 Preview
- Update SQL Server Management Studio 21 Preview
- Uninstall or remove SQL Server Management Studio 21 Preview
- Troubleshoot installation and upgrade issues for SQL Server Management Studio 21 Preview

Feedback

Was this page helpful? Yes No

Troubleshoot installation and upgrade issues for SQL Server Management Studio 21 Preview

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

This troubleshooting guide includes step-by-step instructions to resolve common issues with SQL Server Management Studio (SSMS) installation and upgrade.

Online installations or updates

Try the following possible resolutions that apply to a typical online installation issue in order.

1. Check whether the problem is a known issue

The first thing to check is whether or not the issue you're encountering is a known issue with the Visual Studio Installer that Microsoft is working on fixing.

To see if there's a workaround for your problem, check Known issues in SQL Server Management Studio 21 Preview.

2. Try repairing SSMS

Try to repair your installation. It might fix many common update issues.

3. Delete the installer folder (update issues)

If you encounter an issue when updating, try deleting the Visual Studio Installer folder, and then rerunning the installation bootstrapper. Doing so reinstalls the Visual Studio Installer files and resets the installation metadata.

1. Close the Visual Studio Installer.
2. Delete the Visual Studio Installer folder. Typically, the folder path is `C:\Program Files (x86)\Microsoft Visual Studio\Installer`.

3. Run the SSMS bootstrapper. You might find the bootstrapper in your *Downloads* folder with a file name `vs_SSMS.exe`. Or, you can download the bootstrapper from the installation page for SSMS. Then, run the executable to reset your installation metadata.

4. Try to install or update SSMS again. If the installer continues to fail, report the problem to support.

4. Report the problem to support

In some situations, when there are corrupted files, issues might require case-by-case troubleshooting.

Follow these steps to submit the problem to Microsoft Support:

1. Collect your setup logs. See Collect installation logs for Microsoft Support for details.

2. Open the Visual Studio Installer, and then choose **Report a problem** to open the Feedback tool.

3. This step opens the Developer Community site, and requires you to sign in.

4. Give your problem report a title, and provide the relevant details. See Collect installation logs for Microsoft Support to capture the installation logs for the Visual Studio Installer, and add them as an attachment to the problem report.

5. Choose **Submit**.

5. Remove all SSMS installation files

As a last resort, you can remove all SSMS installation files and product information:

1. Remove all with InstallCleanup.exe.

2. Rerun the SSMS bootstrapper. You might find the bootstrapper in your Downloads folder with a file name `vs_SSMS.exe`. Or, you can download the bootstrapper from the installation page.

3. Try to reinstall SSMS.

Network layout or offline installations

To resolve issues with a network installation, see Error Codes or Troubleshoot network-related errors when you install or use Visual Studio.

Collect installation logs for Microsoft Support

If you contact Microsoft Support, you might be asked to collect setup logs by using the Microsoft Visual Studio and .NET Framework log collection tool. The log collection tool collects setup logs from all components installed by SSMS and Visual Studio, including .NET Framework. The collection tool also collects computer information, a Windows Installer inventory, and Windows event log information for the Visual Studio Installer, Windows Installer, and System Restore.

To collect the logs, follow these steps:

1. Download the tool.

https://aka.ms/vscollect

1. Open an administrative command prompt.
2. Run `collect.exe` in the folder where you saved the tool.

The tool must be run under the same user account that the failed installation was run under. If you're running the tool from a different user account, set the `-user:<name>` option to specify the user account under which the failed installation was run. Run `collect.exe -?` from an administrator command prompt for more options and usage information.

The tool generates a `vslogs.zip` file in your `%TEMP%` folder, typically at `C:\Users\YourName\AppData\Local\Temp\vslogs.zip`.

Report product issues to us using the Report a Problem tool that appears both in the Visual Studio Installer and in the SSMS IDE.

Feedback

Was this page helpful? Yes No

Uninstall or remove SQL Server Management Studio 21 Preview

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

This article walks you through uninstalling SQL Server Management Studio 21 Preview.

> **Tip**
>
> If you're having trouble with your instance of SQL Server Management Studio (SSMS), try the **repair** option.
>
> To change the location for some of your SSMS files, it's possible to do so without uninstalling your current instance as described in Select the installation locations.

Uninstall by using Visual Studio Installer

You can use the Visual Studio Installer can uninstall versions of SSMS.

1. Find the **Visual Studio Installer** on your computer.
2. In the Windows Start menu, you can search for installer.

   > **Note**
   >
   > You can also find the Visual Studio Installer in the following location:
   >
   > `C:\Program Files (x86)\Microsoft Visual Studio\Installer\setup.exe`

   You might have to update the installer before you continue. If so, follow the prompts.
3. In the installer, look for the version of SSMS that you installed. Next, choose **More**, and then choose **Uninstall**.
4. Select **OK** to confirm your choice.

If you change your mind later and want to reinstall SSMS, start the Visual Studio Installer again, choose the **Available** tab, choose SQL Server Management Studio, and then select

Install.

Uninstall SSMS

To remove all installations of SSMS, and the Visual Studio Installer from your machine, uninstall it from **Installed apps**.

1. In Windows 10 and later versions, type **Add or remove programs** in the Windows search box.
2. Find **SQL Server Management Studio**.
3. Choose **Uninstall**.
4. Repeat these steps for all versions of SSMS installed on the machine.
5. Then, find **Microsoft Visual Studio Installer**.
6. Choose **Uninstall**.

Remove all with InstallCleanup.exe

If you experience a catastrophic error and *can't repair or uninstall SSMS*, you can run the `InstallCleanup.exe` tool to remove installation files and product information for all installed instances of SSMS and Visual Studio.

> **⚠ Warning**
>
> Use the InstallCleanup tool only *as a last resort* if repair or uninstall fail. This tool might uninstall features from other Visual Studio installations or other products, which you then might need to repair or reinstall.

Here's how to run the `InstallCleanup.exe` tool:

1. Close the Visual Studio Installer.

2. Open an administrator command prompt. To open an administrator command prompt, follow these steps:
   a. Type `cmd` in the Windows search box.
   b. Right-click **Command Prompt**, and then choose **Run as administrator**.

3. Enter the full path of the `InstallCleanup.exe` tool and add the command-line parameters that you prefer. By default, the path of the tool is as follows. The double quotes enclose a command that contains spaces:

   ```
   Windows Command Prompt
   ```

```
"C:\Program Files (x86)\Microsoft Visual
Studio\Installer\InstallCleanup.exe"
```

> **Note**
>
> The Visual Studio Installer directory is always located at `%ProgramFiles(x86)%\Microsoft Visual Studio`. If you can't find `InstallCleanup.exe` there, follow the instructions to install SSMS. Then, when the workload selection screen is displayed, close the window and follow the steps in this section again.

These options are available:

Expand table

Parameter	Behavior
`-i [version]`	If you don't specify a value, uses the default version. Removes only the main installation directory and product information. Use this parameter if you intend to reinstall the same version of SSMS. If you specify a `[version]` value, the tool removes only products with a version that start with this string value. For example, use the value `-i 21` with to remove all products of version 21.
`-f`	Removes the main installation directory, product information, and most other features installed outside the installation directory that might also be shared with other Visual Studio installations or other products. Use this parameter if you intend to remove SSMS and Visual Studio without reinstalling it later.

Support or troubleshooting

Sometimes, things can go wrong. If your SSMS installation fails, see Troubleshoot installation and upgrade issues for SQL Server Management Studio 21 Preview for step-by-step guidance.

You can reference the SSMS Developer Community site for other known issues (filter on **Tooling** under **Groups**) and to provide feedback to the product team.

Related content

- Modify SQL Server Management Studio 21 Preview components and language packs

- Update SQL Server Management Studio 21 Preview
- Install SQL Server Management Studio 21 Preview

Feedback

Was this page helpful? Yes No

Channels for SQL Server Management Studio 21 Preview

Article • 02/13/2025

SQL Server Management Studio (SSMS) follows a regular release cadence to ensure users have access to the latest features and improvements. This document outlines the different channels available for SSMS, expected release cadence, and how you can provide feedback on new features and updates.

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

We continually enhance the capabilities of SQL Server Management Studio with regular minor version updates according to our roadmap of upcoming features. We introduce these new features and bug fixes first on the **Preview channel** so that you can provide early feedback. When these new features are ready for widespread use, we release them on the **Release channel**. We might sometimes include **Preview Features** to the **Release channel**, which give you an opportunity to try out new features and share your feedback without doing another install.

This article describes our channels for releasing previews, preview features, general releases, and servicing fixes, and how you can provide feedback to us on the features and fixes we release.

1. Preview channel

The Release channel will be available when SQL Server Management Studio 21 is generally available (GA).

Determine your product version and channel

You can determine which channel and version of SSMS you're using by opening **Help** -> **About**. The version number indicates the major version of SSMS by using the first number ("`21`"), the minor update version by using the second number ("21.`0`"), and the build number by using the third number ("21.0.`47`"). If you're using the Preview channel, the preview release is also indicated after the version number ("21.0 **Preview**"), and within SSMS, **PREVIEW** appears near the top right of the SSMS window.

Installing, updating, and switching channels

You might select a channel by either installing a release of SSMS from that channel or, in some cases, by configuring the source location for updates. The following sections link to the installation location for each channel.

You can install multiple channels side-by-side on the same machine. For example, you might install and use Preview channel side-by-side with Release channel. Some shared components, for example Windows SDK, are used by all channels on a machine and are updated to whichever channel has the latest version.

Updates are delivered independently to each channel, and you'll receive a separate notification within each installed version of SSMS if there's an update available for its channel. You can also update each channel by using the Visual Studio Installer.

Release channel

Release channel keeps you up to date with the features ready for widespread mainstream production use. The updates released to the Release channel fall into the following two categories:

- **Minor updates** ship to the Release channel after being available in the Preview channel. These updates might include new features, bug fixes, and changes to adapt to platform updates (for example, changes in Windows or Azure).
- **Servicing updates** are releases of targeted fixes for critical quality, performance, reliability, or security issues.

We stop servicing a minor version update on the Release channel when the next minor version update is released. For example, we won't provide additional servicing updates to version 21.0 once 21.1 has released.

We announce minor updates through the SQL Server blog. All minor and servicing updates are accompanied by release notes. SSMS also alerts you that a new update is available from the notification icon in the bottom right corner of the application, by an entry in the notification hub, and by text in the Visual Studio Installer.

When you update your install of SSMS on the Release channel, you receive the latest servicing update for the latest available minor update.

Preview channel

Preview channel is meant for customers who are eager to try out the latest SSMS features and fixes. It provides you with an early peek of what's coming in the next

version update of the Release channel. You can learn about the features and fixes in the release notes. We also announce the availability of a new Preview release through the SQL Server blog.

The number of preview releases shipped from the Preview channel, before new features ship to the Release channel, varies. Sometimes we also release servicing updates to the Preview channel. You can install both Preview and Release channel versions of SSMS side-by-side on the same machine. At any given point, the Preview channel contains the same or newer features and fixes compared to the Release channel. SSMS notifies you as newer versions are available for install.

Even though previews aren't intended for use in production, they are at a sufficient quality level for you to generally use and provide feedback. We service only the latest preview release. For more information, see SQL Server Management Studio (SSMS) support policy.

Preview features

Although the Preview channel is recommended for trying out new features, we also encourage users in the Current channel to try out new features via **Preview Features**. Preview Features differ from the Preview channel in that each specific feature can be turned on or off. The Preview Features dialog includes description for each feature to help you determine if the feature is relevant to you. Additionally, we hope that you provide feedback on these features so that we can improve them before they're implemented for all customers.

You can turn Preview Features on or off by going to **Tools** > **Options** > **Environment** > **Preview Features**. We sometimes also run experiments with Preview Features, turning them on for a portion of customers to compare the impact against a control group.

Update notifications

You'll receive notification of updates to the currently running SSMS through the notification icon in the application, through posts to the SQL Server blog, and from the Visual Studio Installer. The release notes document the features and fixes available in that release to help you make an informed decision about when to install it.

How to provide feedback

You can report a problem or suggest a feature from **Help** > **Send Feedback**.

1. Open SQL Server Management Studio.
2. Select **Help** > **Send Feedback**.
3. Within the SQL feedback site, provide a title and description for the issue or request. When reporting defects, include detailed steps to recreate the problem.

Support or troubleshooting

Sometimes, things can go wrong. If your SSMS installation fails, see Troubleshoot installation and upgrade issues for SQL Server Management Studio 21 Preview for step-by-step guidance.

You can reference the SSMS Developer Community site for other known issues (filter on **Tooling** under **Groups**) and to provide feedback to the product team.

Related content

- Install SQL Server Management Studio 21 Preview
- Update SQL Server Management Studio 21 Preview
- Modify SQL Server Management Studio 21 Preview components and language packs
- Uninstall or remove SQL Server Management Studio 21 Preview

---

Feedback

Was this page helpful? Yes No

Suggest a feature or improvement for SQL Server Management Studio 21 Preview

Article • 02/13/2025

Users of SQL Server Management Studio (SSMS) can submit requests for improvement or new functionality using the SSMS user feedback site, or using **Help** > **Send feedback**. Submitted ideas are regularly reviewed, and SSMS users can upvote and comment on ideas they support.

When providing feedback, searching the site first is recommended to see if a similar idea exists. Expand the **Groups** menu on the feedback site and select **Tooling**. To search existing items, enter relevant text or keywords in the Search dialog.

Idea status

After you submit a suggestion for a feature or improvement, the Stage indicates where the submission is in its lifecycle. As we consider your feedback and move it along the workflow, we tag it with the corresponding stage. The various stages associated with feature suggestions are listed here, along with a description of their meaning.

Expand table

Stage	Description
Archived	A status of **Archived** means the suggestion is no longer relevant to the current release, or there needs to be more community interest to keep the item active.
Completed	**Completed** indicates that your new feature suggestion is available in the latest product update.
Declined	A status of **Declined** means the suggestion doesn't match our overall product direction and is no longer under consideration.
Needs More Information	A feature suggestion marked **Needs More Information** means that we need more details so we can better understand your suggestion. Check the company response, where we ask for additional information to understand the request better.
New	**New** means the suggestion is newly reported by you or someone else. Action still needs to be taken.

Stage	Description
Planned	A status of **Planned** means that your feature suggestion has a broad community impact and can improve the product experience. We have the item on our roadmap and the feedback item is updated when release information is available.
Under Review	**Under Review** indicates that the feature suggestion is in the queue for prioritization. Considering the product roadmap, we prioritize features to bring the best value to the broader community. Even if we don't implement your new feature suggestion immediately, we'll continue to monitor your idea, let the community weigh in, and then decide on the next steps.

Related content

- Report a problem with SQL Server Management Studio 21 Preview

Feedback

Was this page helpful? Yes No

Report a problem with SQL Server Management Studio 21 Preview

Article • 02/28/2025

You can report a problem from SQL Server Management Studio (SSMS) using the **Help** > **Send feedback** menu or by using the SSMS user feedback site. When reporting an issue, it's recommended that you search the site first to see if a similar issue exists. Expand the **Groups** menu on the feedback site and select **Tooling**. Enter relevant text or keywords in the Search dialog to find existing items for the problem.

Include as much detail as possible when describing the problem, including step-by-step instructions and information such as the version of SSMS (available in **Help** > **About**), the version of SQL, and the Operating System version and language. For any error message, include the entire text and use the **Show Details** button in the error dialog to capture complete details about the error to add to the item description.

Idea status

After you report a problem, the Stage indicates where the issue is in its lifecycle. As we review the issue and move it along the workflow, we tag it with the corresponding stage. The various stages associated with reported ideas are listed here, along with a description of their meaning.

Expand table

Stage	Description
Archived	A status of **Archived** means the problem is no longer applicable or insufficient information was provided to recreate the problem.
Completed	**Completed** indicates that the latest product update fixes the problem.
Needs More Information	Any problem marked **Needs More Information** means we need more details to recreate the problem internally. Check the company response, where we ask for additional information about the problem.
New	**New** means the problem is newly reported by you or someone else. Action still needs to be taken.
Planned	A status of **Planned** means that we recreated the problem, logged a tracking item internally, and planned to fix the problem in a future release.

Related content

- Suggest a feature or improvement for SQL Server Management Studio 21 Preview

Feedback

Was this page helpful? Yes No

The Customer Experience Improvement Program

Article • 02/13/2025

As a Visual Studio-based application, SQL Server Management Studio (SSMS) is part of the Visual Studio Customer Experience Improvement Program (VSCEIP). The program is designed to help Microsoft improve SSMS and Visual Studio over time. This program collects information about errors, computer hardware, and how people use both applications without interrupting users in their tasks at the computer. The information collected helps Microsoft identify which features need improvement. This document covers how to opt in or out of the VSCEIP and provides information about the types of data we collect and how we use it.

Opt out of diagnostic data collection

Given the purpose of the data we collect and the constraints on its access and retention, we recommend that you use the default privacy settings for SQL Server Management Studio and Windows. You can opt out of the program, but you're opting out of optional diagnostic data collection. Some diagnostic data collection is required to ensure SSMS is secure, up to date, and performing as expected. Required diagnostic data collection isn't affected by your choice to opt out of VSCEIP.

Opt in or out

The VSCEIP is turned on by default. You can turn it off or back on again by following these instructions:

1. In SSMS select **Help** > **Privacy** > **Privacy Settings**.
2. The **Visual Studio Experience Improvement Program** dialog opens.
3. To opt out, select **No, I would not like to participate**, and then select OK.
4. To opt in, select **Yes, I am willing to participate (Recommended)**, and then select OK.

Related content

- Suggest a feature or improvement for SQL Server Management Studio 21 Preview

- Report a problem with SQL Server Management Studio 21 Preview

Feedback

Was this page helpful? Yes No

User permissions and SQL Server Management Studio 21 Preview

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

For reasons of security, you should run SQL Server Management Studio (SSMS) as a typical user whenever possible.

> **⚠ Warning**
>
> Be sure not to launch or run any SSMS solution that doesn't come from a trusted person or a trusted location.

You can do nearly everything in the SSMS application as a typical user. You need administrator permissions to complete the following tasks:

Expand table

Area	Task	For more information
Installation	Initial installation and configuration of SSMS	Install SQL Server Management Studio
Installation	Modify or update SSMS. Requires administrator permissions by default	Update SQL Server Management Studio, Modify SQL Server Management Studio
Help	Install, update, or remove local Help content.	Install and manage local Help content

Run SSMS as an administrator

If you need to run SSMS as an administrator, here's how.

Use the Start menu

1. Depending on the version of Windows you're using, perform one of the following steps:

- In **Windows 10**, open the **Start** menu, and then scroll to **SQL Server Management Studio 21 Preview**.
- In **Windows 11**, select the **Start** button, and then in the **Search** box, type **SQL Server Management Studio 21 Preview**.

2. Next, right-click **SQL Server Management Studio 21 Preview**, and then select **Run as administrator**.

   When SSMS starts, **(Administrator)** appears after the product name in the title bar, and **ADMIN** appears near the top right of the SSMS window, in the same area where **PREVIEW** appears if you're using a preview build of SSMS. You can select this button to view the **About** dialog.

Modify the shortcut

You can also modify the application shortcut to always run with administrative permissions. Here's how.

Windows 10

1. Open the **Start** menu, scroll to **SQL Server Management Studio 21 Preview**, and then select **More** > **Open file location**.
2. In **File Explorer**, locate the **SQL Server Management Studio 21 Preview** shortcut for the version that you're using. Then, right-click the shortcut and select **Send to** > **Desktop (create shortcut)**.
3. On the **Windows 10** desktop, right-click the **SQL Server Management Studio 21 Preview** shortcut, and then select **Properties**.
4. Select the **Advanced** button, and then select the **Run as administrator** check box.
5. Select **OK**, and then select **OK** again.

Support or troubleshooting

Sometimes, things can go wrong. If your SSMS installation fails, see Troubleshoot installation and upgrade issues for SQL Server Management Studio 21 Preview for step-by-step guidance.

You can reference the SSMS Developer Community site for other known issues (filter on **Tooling** under **Groups**) and to provide feedback to the product team.

Related content

- Install SQL Server Management Studio 21 Preview
- Update SQL Server Management Studio 21 Preview
- Modify SQL Server Management Studio 21 Preview components and language packs
- Uninstall or remove SQL Server Management Studio 21 Preview

---

Feedback

Was this page helpful? Yes No

Sign in or switch user accounts in SQL Server Management Studio 21 Preview

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

In this article, you learn how to sign in to SQL Server Management Studio (SSMS), add and switch user accounts, update your profile, sign out of your account, and the benefits to signing in. To learn how to access multiple user accounts in SSMS, see Access multiple accounts. For guidance on how to add your public GitHub or GitHub Enterprise account to your SSMS keychain, see Work with GitHub accounts in Visual Studio.

Sign in to your account

When you sign in to SSMS, you can access your Azure account and synchronize your settings across devices.

To add another account, see Add and switch user accounts in SSMS.

1. Launch SSMS. When you open SSMS for the first time, you're prompted to sign in and provide some basic registration information.

   > **Note**
   >
   > Sign in isn't required to use SSMS. If you choose to not sign in when you first open SSMS, it's easy to do so later. Look for the **Sign in** link in the upper-right corner of the SSMS environment.

2. Choose a Microsoft account or a work or school account. If you don't have one, you can create a Microsoft account for free.

You can see that you've successfully signed in, in the upper-right corner of the SSMS environment. Unless you sign out, you automatically sign in to SSMS whenever you start it, and any changes to synchronized settings are automatically applied.

Sign in from the SSMS application

If you choose to not sign in when you first open SSMS, you can sign in later.

1. Select the **Sign in** icon in the upper-right corner of the SSMS environment, and then choose an account to sign into.
2. On the **Sign in** page, select a connected account under **Use one of these accounts**, or select **Use another account** to sign in with another account.
3. Follow the prompts to enter the new account credentials.

Benefits: why sign in?

While you don't have to sign in, there are many advantages to doing so.

Expand table

Benefit	Description
Auto-connect to Azure	Access to your Azure account in SSMS without being prompted again for credentials for the same account.
Synchronize your settings	Settings that you customize, such as key bindings, window layout, and color theme, apply immediately when you sign in to SSMS on any device.

Add and switch user accounts in SSMS

If you have multiple accounts, you can add them all to SSMS so that you can access the resources from any account without having to sign in to it separately.

After you add multiple accounts on one machine, that group of accounts roams with you if you sign in to SSMS on another machine. However, if your account credentials don't roam, you'll be prompted to enter credentials for those other accounts the first time you attempt to use the resources on a new machine.

To authenticate and access Azure resources from SSMS, sign in to SSMS with an account that has access to Azure resources.

There are several types of accounts you can add:

- Add a standard Azure account:

    1. Select the icon with your profile name in the upper-right corner of the SSMS environment.
    2. Select **Add another account** and then choose an account to sign into.
    3. On the **Sign in** page, select the account or choose **Use another account**.
    4. Follow the prompts to enter the new account credentials.

- Add a GitHub account
- Add a Multifactor authentication (MFA) account

Remove an account

You can remove any of the accounts you've added to SSMS. When you remove an account, only that specific account is affected and any resources associated with that account are no longer accessible. However, if you select **Sign out**, all accounts are signed out.

1. Select the icon with your profile name in the upper-right corner of your SSMS environment.
2. Select **Account Settings** to view the list of accounts signed in to SSMS.
3. Select **Remove account** for the account you want to remove.
4. Confirm the removal on the dialog that appears.

Sign out of all accounts

Signing out of SSMS removes all accounts, settings no longer roam across devices, and associated resources aren't accessible anymore.

To sign out of all accounts:

1. Select the icon with your profile name in the upper-right corner of your SSMS environment.
2. Select **Sign out**.

You can also use the **Account Settings** dialog to sign out of all accounts.

Related content

- Access multiple accounts in SQL Server Management Studio 21 Preview
- Work with GitHub accounts in Visual Studio
- Sign in to Visual Studio with accounts that require multifactor authentication (MFA)

Feedback

Was this page helpful? Yes No

Access multiple accounts in SQL Server Management Studio 21 Preview

Article • 02/13/2025

In this article, you learn how to access multiple accounts in SQL Server Management Studio (SSMS). After you sign in to SSMS with a Microsoft or an organizational account, you can see the resources accessible from your accounts in places such as the **Azure Storage** dialog or the **Add Azure firewall rule** dialog.

Access your Azure account via the Azure Storage dialog

1. Open Object Explorer using **View** -> **Object Explorer**.
2. From the **Connect** dropdown list, select **Azure Storage**.
3. In the **Connect to a Microsoft subscription** dialog, select the **Change User** button.
4. In the **Select a Microsoft account** dialog, use the dropdown list to select a different account, or do add a new account.
5. Once you have selected an account, select **OK**.

   The available values for **Select Storage Account** and **Select Blob Container** updates are based on the account selected.

Related content

- Sign in or switch user accounts in SQL Server Management Studio 21 Preview

---

Feedback

Was this page helpful? Yes No

Release history for SQL Server Management Studio

Article • 02/13/2025

SQL Server Management Studio is updated regularly to provide new features, performance and reliability improvements, security updates, and bug fixes.

With SQL Server Management Studio 21 Preview, we introduce Preview and Release channels to provide customers the flexibility and control to determine when they adopt feature updates. The latest release on each channel is always, by definition, the most secure, so we highly encourage customers to stay up to date regardless of what channel they choose to get updates from. Refer to our support policy or our channels documentation for additional information.

Release notes

Here are the release notes for different versions of SQL Server Management Studio:

- SQL Server Management Studio 21 Preview release notes
- SQL Server Management Studio 20 release notes

Update your installation to a specific release

If you need to install or update to a specific release of SSMS, you can find the available bootstrapper in the Evergreen bootstrapper] table.

Uninstall SSMS to go back to an earlier release

If you need to install an earlier release of SQL Server Management Studio 21, you must uninstall your current installation and use one of the links below to reinstall the version that you prefer.

1. On Windows, open the Visual Studio Installer.
2. Uninstall all instances of SQL Server Management Studio 21 listed in the Visual Studio Installer.
3. From Programs and Features in the Control Panel, find **SQL Server Management Studio 21**, and uninstall it.

If you need to install an earlier release of SQL Server Management Studio 20.x and earlier for a major version (for example, to downgrade from SSMS 20.2 to SSMS 20.1), then you must uninstall your current installation and use one of the links below to reinstall the version that you prefer.

1. On Windows, from Programs and Features in the Control Panel, find your version of **Microsoft SQL Server Management Studio** and uninstall it.

Before attempting to install a previous version of SQL Server Management Studio, refer to our support policy. Microsoft doesn't guarantee support outside of this policy.

If you're unable to follow the steps above due to a corrupted install, follow the steps to remove SQL Server Management Studio with the InstallCleanup.exe tool.

Uninstalling SSMS won't remove standalone products and components such as .NET and Visual C++ Redistributables. These might need to be removed manually from **Apps and Features** in Windows Settings.

Release dates and build numbers

This section provides links to the publicly released bootstrappers for SQL Server Management Studio. The bootstrappers initiate Visual Studio's installation process. Only the latest release is supported.

To verify what version a given bootstrapper installs, right-click on the bootstrapper `vs_ssms.exe`, choose **Properties** > **Details**, and look at the Product Version information. The version number should always be read as "latest of that version". A bootstrapper with a Product Version of 21.0 would install the latest 21.0.x servicing release that is available.

Evergreen bootstrapper

The bootstrapper in the following table will always install the latest release of SSMS from its respective channel.

Expand table

Channel	Version	Bootstrapper	End of Support date
Preview	21.0	SQL Server Management Studio 21 Preview	

Fixed version installers

This table lists latest build numbers of publicly released versions of SQL Server Management Studio, the release date, and download link.

Expand table

SSMS version	Build number	Release date
20.2	20.2.30.0	July 9, 2024
19.3	19.3.4.0	January 10, 2024
18.12.1	15.0.18420.0	June 21, 2022
17.9.1	14.0.17289.0	November 21, 2018
16.5.3	13.0.16106.4	January 30, 2017

Related content

- Channels for SQL Server Management Studio 21 Preview
- SQL Server Management Studio (SSMS) support policy

Feedback

Was this page helpful? Yes No

SQL Server Management Studio 21 Preview product roadmap

Article • 02/13/2025

SQL Server Management Studio 21 Preview continues to be an integrated environment for managing any SQL infrastructure, which can be used to access, configure, manage, administer, and develop all components of SQL Server, Azure SQL Database, Azure SQL Managed Instance, SQL Server on Azure VM, and Azure Synapse Analytics. SSMS provides a single comprehensive application that combines a broad group of graphical tools with a rich script editor to provide access to SQL Server and Azure SQL databases for data professionals, database administrators, and database developers of all skill levels.

This roadmap describes the work planned for subsequent SQL Server Management Studio 21 Preview releases, and encompasses themes including modernization, AI, Git integration, and general improvements across the application.

Modernization

SQL Server Management Studio 21 Preview introduces updates that significantly enhance the user experience and functionality. These improvements include updating to Visual Studio 2022 which brings support for 64-bit, installation using the Visual Studio Installer, support for automatic updates, and initial support for dark theme. Full support for dark theme across all dialogs will take numerous releases to fully implement. Initial support exists for dialogs that are most frequently accessed by users, including Object Explorer, Query Editor, Results pane, and Template Explorer. If you plan to use dark theme, and would like to provide feedback on the dialogs that you use the most, please go to the SSMS 21 dark theme support feedback item and add a comment.

AI with Copilot

AI with Copilot in SSMS brings a sidecar chat associated with the Query Editor window from which it launches. Copilot has both server and database context, based on the connection. This feature allows users to ask SQL questions and get assistance writing, editing, and fixing Transact-SQL using natural language prompts. Copilot in SSMS will be an optional installation, available as an extension. Initially, Copilot in SSMS is only available in preview.

Extensibility and personalization

Extensibility in SQL Server Management Studio 21 Preview starts with parity of what is available in the current release of SSMS. Third-party extensions SQL Server Management Studio 21 Preview can be installed manually from an exe or msi file. Existing extensions might not work with SSMS 21 until they're modified to have 64-bit support, and might require additional updates. Over time, we plan to provide full support for extensions, including the ability to browse extensions from within SSMS, a presence in the Visual Studio marketplace, and support of the VSIX Installer.

Git tooling

Git integration returns to SSMS, allowing administrators and database developers to manage script repositories. This integration supports common Git operations such as cloning, committing, pushing, and pulling changes directly from SSMS. You can create internal GitHub repos with better guardrails to ensure security, providing a seamless and secure way to manage your codebase.

Performance and productivity

SQL Server Management Studio 21 Preview also includes performance and productivity improvements:

- A new connection experience that simplifies the process of connecting to SQL Server instances and Azure SQL databases.
- Start time improvements, reducing the time it takes to launch SSMS and get to work.

Query Editor

The Query Editor experience is enhanced with new features, including support for vertical tabs, sorting of tabs, setting minimum and maximum widths for tabs, and coloring tabs based on project or file extension.

Security

Azure support continues to improve with the adoption of the Azure authentication methods used by Visual Studio, ultimately rolling out to all Azure integration dialogs. With SQL Server Management Studio 21 Preview, you can create logins and users for

Azure SQL Database with the UI. New security features in SQL Server Management Studio 21 Preview include an Always Encrypted Assessment wizard, to help determine which columns in a database are candidates for encryption.

Feedback

Was this page helpful? Yes No

Download SQL Server Management Studio (SSMS)

Article • 04/08/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics SQL analytics endpoint in Microsoft Fabric Warehouse in Microsoft Fabric

SQL Server Management Studio (SSMS) is an integrated environment for managing any SQL infrastructure, from SQL Server to Azure SQL Database. SSMS provides tools to configure, monitor, and administer instances of SQL Server and databases. Use SSMS to deploy, monitor, and upgrade the data-tier components used by your applications and build queries and scripts.

Use SSMS to query, design, and manage your databases and data warehouses, wherever they are - on your local computer or in the cloud.

For customers needing a cross-platform companion to SSMS for managing SQL and other Azure databases, use Azure Data Studio.

For details and more information about what's new in this release, see Release notes for SQL Server Management Studio (SSMS) 20.2.1.

For details and download information for SQL Server Management Studio 21 Preview 1, see Install SQL Server Management Studio 21 Preview

Download SSMS

Download SQL Server Management Studio (SSMS) 20.2.1

SSMS 20.2.1 is the latest generally available (GA) version. If you have a *preview* version of SSMS 20 installed, uninstall it before installing SSMS 20.2.1. Installing SSMS 20.2.1 doesn't upgrade or replace SSMS 19.x and earlier versions.

- Release number: 20.2.1
- Build number: 20.2.37.0
- Release date: April 8, 2025

By using SQL Server Management Studio, you agree to its license terms and privacy statement. If you have comments or suggestions or want to report issues, the best way to contact the SSMS team is at the SSMS Developer Community site.

The SSMS 20.x installation doesn't upgrade or replace SSMS 19.x and earlier versions. SSMS 20.x installs alongside previous versions, so both versions are available. However, if you have an

earlier *preview* version of SSMS 20 installed, you must uninstall it before installing the latest release of SSMS 20. You can see if you have a preview version by going to the **Help** > **About** window.

If a computer contains side-by-side installations of SSMS, verify you start the correct version for your specific needs. The latest version is labeled **Microsoft SQL Server Management Studio v20.2.1**.

> **Important**
>
> For SQL Server Management Studio (SSMS) 18.7 through 19.3, Azure Data Studio is automatically installed alongside SSMS. Users of SQL Server Management Studio are able to benefit from the innovations and features in Azure Data Studio. Azure Data Studio is a cross-platform and open-source desktop tool for your environments, whether in the cloud, on-premises, or hybrid.
>
> To learn more about Azure Data Studio, check out What is Azure Data Studio or the FAQ.

Available languages

This release of SSMS can be installed in the following languages:

SQL Server Management Studio 20.2.1:

- Chinese (Simplified)
- Chinese (Traditional)
- English (United States)
- French
- German
- Italian
- Japanese
- Korean
- Portuguese (Brazil)
- Russian
- Spanish

If you access this page from a non-English language version and want to see the most up-to-date content, select **Read in English** at the top of this page. To download different languages, select available languages.

> **Note**

The SQL Server PowerShell module is a separate install through the PowerShell Gallery. For more information, see Install the SQL Server PowerShell module.

What's new

For details and more information about what's new in this release, see Release notes for SQL Server Management Studio (SSMS).

Previous versions

This article is for the latest version of SSMS only. To download previous versions of SSMS, visit Previous SSMS releases.

> **Note**
>
> In December 2021, releases of SSMS prior to 18.6 will no longer authenticate to Database Engines through Microsoft Entra multifactor authentication (MFA). To continue utilizing MFA, you need SSMS 18.6 or later.
>
> Connectivity to Azure Analysis Services through Microsoft Entra multifactor authentication requires SSMS 18.5.1 or later.

Unattended install

You can install SSMS using PowerShell.

Follow these steps to install SSMS in the background with no GUI prompts.

1. Launch PowerShell with elevated permissions.
2. Type the following command.

```PowerShell
$media_path = "<path where SSMS-Setup-ENU.exe file is located>"
$install_path = "<root location where all SSMS files will be installed>"
$params = " /Install /Quiet SSMSInstallRoot=$install_path"

Start-Process -FilePath $media_path -ArgumentList $params -Wait
```

For example:

PowerShell

```
$media_path = "C:\Installers\SSMS-Setup-ENU.exe"
$install_path = "$env:SystemDrive\SSMSto"
$params = "/Install /Quiet SSMSInstallRoot=`"$install_path`""

Start-Process -FilePath $media_path -ArgumentList $params -Wait
```

You can also pass `/Passive` instead of `/Quiet` to see the setup UI.

3. If all goes well, you can see SSMS installed at *%systemdrive%\SSMSto\Common7\IDE\Ssms.exe* based on the example. If something went wrong, you could inspect the error code returned and review the log file in `%TEMP%\SSMSSetup`.

Uninstall

SSMS might install shared components if it determines they're missing during SSMS installation. SSMS doesn't automatically uninstall these components when you uninstall SSMS.

The shared components are:

- Microsoft OLE DB Driver 18 for SQL Server
- Microsoft ODBC Driver 17 for SQL Server
- Microsoft Visual C++ 2017 Redistributable (x86)
- Microsoft Visual C++ 2017 Redistributable (x64)
- Microsoft Visual Studio Tools for Applications 2019

These components aren't uninstalled because they can be shared with other products. If uninstalled, you might run the risk of disabling other products.

Installation with Azure Data Studio

- SSMS installs Azure Data Studio by default for versions 18.7 through 19.3. SSMS does not install Azure Data Studio with SSMS 20.0 and higher.
  - The installation of Azure Data Studio by SSMS is skipped if an equal or higher version of Azure Data Studio is already installed.
  - The Azure Data Studio version can be found in the release notes.
- The Azure Data Studio system installer requires the same security rights as the SSMS installer.
- The Azure Data Studio installation is completed with the default Azure Data Studio installation options. These are to create a Start Menu folder and add Azure Data Studio to

PATH. A desktop shortcut isn't created, and Azure Data Studio isn't registered as a default editor for any file type.

- Localization of Azure Data Studio is accomplished through Language Pack extensions. Install the corresponding language pack from the extension marketplace to localize Azure Data Studio.
- For versions 18.7 through 19.3, the installation of Azure Data Studio can be skipped by launching the SSMS installer with the command line flag `DoNotInstallAzureDataStudio=1`.

Supported SQL offerings

- This version of SSMS works with SQL Server 2014 (12.x) and later versions. It provides the most significant support for working with the latest cloud features in Azure SQL Database, Azure Synapse Analytics, and Microsoft Fabric.

- Additionally, SSMS 20.x can be installed alongside with SSMS 19.x, SSMS 18.x, SSMS 17.x, and SSMS 16.x.

- For SQL Server Integration Services (SSIS), SSMS 17.x and later versions don't support connecting to the legacy SQL Server Integration Services service. To connect to an earlier version of the legacy Integration Services, use the version of SSMS aligned with the version of SQL Server. For example, use SSMS 16.x to connect to the legacy SQL Server 2016 (13.x) Integration Services service. SSMS 17.x and SSMS 16.x can be installed on the same computer. Since the release of SQL Server 2012 (11.x), the SSIS Catalog database, SSISDB, is the recommended way to store, manage, run, and monitor Integration Services packages. See SSIS Catalog for details.

SSMS system requirements

The current release of SSMS supports the following 64-bit platforms when used with the latest available service pack:

Supported operating systems:

- Windows 11 (x64)
- Windows 10 (x64) version 1607 (10.0.14393) and later versions
- Windows Server 2022 (x64)
- Windows Server Core 2022 (x64)
- Windows Server 2019 (x64)
- Windows Server Core 2019 (x64)
- Windows Server 2016 (x64) [1]

[1] SSMS requires .NET Framework 4.7.2.

Additional requirements:

- Administrator rights are required to install or update SSMS.
- Running in a virtual machine environment requires a full Windows operating system.
- Windows containers are not supported.
- SSMS is not supported in application virtualization solutions such as Microsoft App-V or MSIX for Windows, or third-party app virtualization technologies.

> **Note**
>
> To install SSMS on Windows Server Core, you must install the Server Core App Compatibility Feature on Demand.

Supported hardware:

- 1.8 GHz or faster x86 (Intel, AMD) processor. Dual-core or better recommended
- 2 GB of RAM; 4 GB of RAM recommended (2.5 GB minimum if running on a virtual machine)
- Hard disk space: Minimum of 3 GB up to 10 GB of available space

> **Note**
>
> SSMS is available only as a 32-bit application for Windows. If you need a tool that runs on operating systems other than Windows, we recommend Azure Data Studio. Visit Azure Data Studio, for more details.

Get help for SQL tools

- All the ways to get help
- SQL Server product feedback
- SSMS user feedback
- SQL Server on Microsoft Q & A
- Support options for business users

Related content

- SQL tools overview
- What is SQL Server Management Studio (SSMS)?
- What is Azure Data Studio?
- Download SQL Server Data Tools (SSDT) for Visual Studio

- Latest updates and version history for SQL Server
- Azure Data Architecture Guide
- SQL Server Blog

Contribute to SQL documentation

Did you know that you can edit SQL content yourself? If you do so, not only do you help improve our documentation, but you also get credited as a contributor to the page.

For more information, see Edit Microsoft Learn documentation.

Release notes for SQL Server Management Studio (SSMS)

Article • 04/08/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

This article details updates, improvements, and bug fixes for the current and previous versions of SSMS.

> **Note**
>
> In December 2021, releases of SSMS prior to 18.6 will no longer authenticate to Database Engines through Microsoft Entra multifactor authentication (MFA). To continue utilizing MFA, you need SSMS 18.6 or later.
>
> Connectivity to Azure Analysis Services through Microsoft Entra multifactor authentication requires SSMS 18.5.1 or later.

Current SSMS release

Download SQL Server Management Studio (SSMS) 20.2.1

SSMS 20.2.1 is the latest general availability (GA) release of SSMS. If you need a previous version of SSMS, see previous SSMS releases.

20.2.1

Download SSMS 20.2.1

- Release number: 20.2.1
- Build number: 20.2.37.0
- Release date: April 8, 2025

Available languages:

- Chinese (Simplified)
- Chinese (Traditional)
- English (United States)
- French
- German

- Italian
- Japanese
- Korean
- Portuguese (Brazil)
- Russian
- Spanish

Security advisories 20.2.1

Expand table

Feature	Details
CVE-2025-29803	This security update addresses an Elevation of Privilege vulnerability in the Visual Studio Tools for Applications 2022 Installer.

Known issues 20.2.1

Expand table

Feature	Details	Workaround
Analysis Services	When you connect to Analysis Services with Microsoft Entra MFA, if you add a new role or open properties for a role, the message "the identity of the user being added to the role is not fetched properly" appears.	This error is benign and can be ignored. The error is addressed within the Azure infrastructure, and no updates to SSMS are required.
Analysis Services	After adding a new role, or when opening properties for an existing role, you can't use **Search by name or email address** to add a user.	A user can be added with the **Manual Entry** option.
Database Designer	Selecting the Design option for a view referencing a table using spatial data causes SSMS to crash.	Use T-SQL to make changes to the view.
Database Mirroring	If you launch the Database Mirroring Monitor from the mirrored node, the primary node isn't listed.	Register the mirrored node from Database Mirroring Monitoring, or use SSMS 18.12.1 to monitor from the mirrored node.
General SSMS	Import settings from SSMS 17 option not available.	Settings can be imported from SSMS 18.

Feature	Details	Workaround
Linked servers	Creating a linked server to Azure SQL Database with SQL Server selected as Server type connects to the `master` database.	To create a linked server to Azure SQL Database, select **Other data source** for the **Server type**, and select **Microsoft OLE DB Provider for SQL Server** or **Microsoft OLE DB Driver for SQL Server** as the **Provider**. Enter the logical server name in the Data source field and the database name in the Catalog field.
Maintenance Plan	Selecting "Contents" after adding a backup file within the Destination pane of the Backup Database Task, causes the dialog to disappear.	Use SSMS 20.1 or SSMS 19.3 to access the Contents dialog.
PolyBase	PolyBase node isn't visible in Object Explorer when you connect to SQL Server 2022 (16.x).	Use SSMS 18.12.1.
Profiler	The Profiler menu isn't localized.	No current alternative.
Replication	If Azure SQL Managed Instance is the publisher and SSMS is running on a machine that isn't in the same virtual network as the publisher, you aren't able to insert a tracer token via Replication Monitor.	To insert tracer tokens, use Replication Monitor in SSMS on a machine that is in the same virtual network as the Azure SQL Managed Instance publisher.
Stretch Database	Removed Stretch Database Wizard.	Use T-SQL to configure Stretch Database or use SSMS 18.9.1 or earlier to use the Stretch Database Wizard.

See Known issues using Strict Encryption in 20.0 for known issues using SSMS 20.x and Strict Encryption.

You can reference the SSMS Developer Community site for other known issues and to provide feedback to the product team.

Previous SSMS releases

Download previous SSMS versions by selecting the download link in the related section.

Expand table

SSMS version	Build number	Release date
20.2	20.2.30.0	July 9, 2024

SSMS version	Build number	Release date
20.1	20.1.10.0	April 9, 2024
20.0	20.0.70.0	March 19, 2024
19.3	19.3.4.0	January 10, 2024
18.12.1	15.0.18420.0	June 21, 2022
17.9.1	14.0.17289.0	November 21, 2018
16.5.3	13.0.16106.4	January 30, 2017

20.2

Download SSMS 20.2

- Release number: 20.2
- Build number: 20.2.30.0
- Release date: July 9, 2024

Available languages:

- Chinese (Simplified)
- Chinese (Traditional)
- English (United States)
- French
- German
- Italian
- Japanese
- Korean
- Portuguese (Brazil)
- Russian
- Spanish

What's new in 20.2

Expand table

Feature	Details
Always Encrypted	Introduced support for temporal tables.
Always Encrypted	Introduced logging for the Always Encrypted wizard to facilitate troubleshooting.

Feature	Details
Drivers	Updated SSMS to use the latest driver version for MSOLEDBSQL (18.7.4). The inclusion of this new version could require users who also have older versions of the driver to reboot after installing SSMS 20.2. For more information, review the release notes for the Microsoft OLE DB driver.
Integration Services	Removed HADOOP files from SQL Server Integration Services (SSIS) installation files, this addresses CVE-2022-25168.
Libraries	Updated DacFx to version 162.3.566.
Libraries	Updated Server Management Objects (SMO) to version 171.36.0.
Libraries	Removed the Microsoft Visual C++ 2013 Redistributable (x86) from the SSMS installation. Upgrading from a previous version of 20.x does not remove the files.
Link feature for Azure SQL Managed Instance	Introduced support for a simplified link failover experience.

Bug fixes in 20.2

Expand table

Feature	Details
Accessibility	Added accessibility support for Expand/Collapse in Database Properties.
Accessibility	Improved accessibility of radio buttons in the Restore Database dialog using arrow or tab keys.
Accessibility	Fixed labels for radio button controls in the Files page of Database Properties.
Accessibility	Fixed an issue with focus control in the Data Classification page.
Accessibility	Addressed issues with incomplete or unnecessary screen reader announcements in the Data Classification page.
Always Encrypted	Fixed error "Object reference not set to an instance of an object", which occurred when trying to create a Column Master Key after signing out of Azure.
Connection	Addressed an issue with truncated authentication methods in the Connection dialog when using Russian locale.
Connection	Fixed incorrect length of dropdown lists after changing the engine type.
Integration Services	Resolved error "The certificate chain was issued by an authority that is not trusted" when creating or modifying an Integration Services job step in SQL Agent. See

Feature	Details
	SSMS 20 - certificate error when viewing or editing Agent jobs that run SSIS packages.
Link feature for Azure SQL Managed Instance	Resolved a problem where SQL Server endpoint certificates were not loaded for the Managed Instance.
Object Explorer	Updated the script generated for external file formats to include the FIRST ROW property.
Object Explorer	Added Table-Valued Functions node within the Programmability > Functions node for Synapse.
Query Editor	Updated lock icons in the query editor toolbar to be color-aware.
Query Editor	Addressed error "Unable to query transaction count. The SQL text editor window will close without committing any open transactions" when closing an unsaved editor with either SHOWPLAN_ALL or SHOWPLAN_XML enabled, and the option **Check for open transactions before closing T-SQL query windows** enabled.
Query Plans	Reduced the number of characters in the Description of an execution plan to 1000. The full query is available using the ellipses.
Query Store	Addressed an issue where the **Queries With Forced Plans** report generated the error "Couldn't connect to database", see Query Store report 'Queries with Forced Plans' fails trying to sort by last execution time.
Query Store	Fixed Tracked Queries report to correctly display the metric selected in the Configure dialog.
Query Store	Removed unnecessary border around options within the Configure dialog.
Query Store	Fixed an issue where no metric was selected in the Configure dialog for the Tracked Queries report.
Query Store	Addressed behavior where drop down menus in the report for Metric and Statistic were not updated after they were changed in the Configure dialog.
Replication	Fixed an issue where Replication Conflict Viewer was inaccessible when using Mandatory or Optional encryption, see SSMS 20.0 (Replication - View Conflicts) bug.

Known issues 20.2

Expand table

Feature	Details	Workaround
Analysis Services	When you connect to Analysis Services with Microsoft Entra MFA, if you add a new role or open properties for a role, the message "the identity of the user being added to the role is not fetched properly" appears.	This error is benign and can be ignored. The error is addressed within the Azure infrastructure, and no updates to SSMS are required.
Analysis Services	After adding a new role, or when opening properties for an existing role, you can't use **Search by name or email address** to add a user.	A user can be added with the **Manual Entry** option.
Database Designer	Selecting the Design option for a view referencing a table using spatial data causes SSMS to crash.	Use T-SQL to make changes to the view.
Database Mirroring	If you launch the Database Mirroring Monitor from the mirrored node, the primary node isn't listed.	Register the mirrored node from Database Mirroring Monitoring, or use SSMS 18.12.1 to monitor from the mirrored node.
General SSMS	Import settings from SSMS 17 option not available.	Settings can be imported from SSMS 18.
Linked servers	Creating a linked server to Azure SQL Database with SQL Server selected as Server type connects to the `master` database.	To create a linked server to Azure SQL Database, select **Other data source** for the **Server type**, and select **Microsoft OLE DB Provider for SQL Server** or **Microsoft OLE DB Driver for SQL Server** as the **Provider**. Enter the logical server name in the Data source field and the database name in the Catalog field.
Maintenance Plan	Selecting "Contents" after adding a backup file within the Destination pane of the Backup Database Task, causes the dialog to disappear.	Use SSMS 20.1 or SSMS 19.3 to access the Contents dialog.
PolyBase	PolyBase node isn't visible in Object Explorer when you connect to SQL Server 2022 (16.x).	Use SSMS 18.12.1.
Profiler	The Profiler menu isn't localized.	No current alternative.
Replication	If Azure SQL Managed Instance is the publisher and SSMS is running on a machine that isn't in the same virtual network as the publisher, you	To insert tracer tokens, use Replication Monitor in SSMS on a machine that is in the same virtual network as the Azure SQL Managed Instance publisher.

Feature	Details	Workaround
	aren't able to insert a tracer token via Replication Monitor.	
Stretch Database	Removed Stretch Database Wizard.	Use T-SQL to configure Stretch Database or use SSMS 18.9.1 or earlier to use the Stretch Database Wizard.

See Known issues using Strict Encryption in 20.0 for known issues using SSMS 20.x and Strict Encryption.

You can reference the SSMS Developer Community site for other known issues and to provide feedback to the product team.

20.1

Download SSMS 20.1

- Release number: 20.1
- Build number: 20.1.10.0
- Release date: April 9, 2024

Available languages:

- Chinese (Simplified)
- Chinese (Traditional)
- English (United States)
- French
- German
- Italian
- Japanese
- Korean
- Portuguese (Brazil)
- Russian
- Spanish

What's new in 20.1

Expand table

Feature	Details
Drivers	Updated SSMS to use the latest driver versions for MSODBCSQL.MSI (17.10.6.1) and MSOLEDBSQL.MSI (18.7.2). The inclusion of these new versions could require users who also have older versions of the drivers to reboot after installing SSMS 20.1. For more information, review the release notes for the Microsoft ODBC driver and the Microsoft OLE DB driver.
Drivers	Updated Microsoft.Data.SqlClient version from 5.1.4 to 5.1.5.
Libraries	Updated Server Management Objects (SMO) version to 171.31.0.
Libraries	Updated Microsoft Visual C++ Redistributable version to 14.38.33135.0. The inclusion of these new versions could require users who also have older versions of the drivers to reboot after installing SSMS 20.1.
Object Explorer	Updated Object Explorer to display table names prefixed with schema in Graph edge constraint connections.

Bug fixes in 20.1

Expand table

Feature	Details
Maintenance Plans	Resolved issue where Backup Database Task dialog closed after selecting the **Files and Filegroups** radio button in the dialog.
SqlParser	Added support for DEFAULT_DATABASE option to CREATE LOGIN T-SQL syntax when using EXTERNAL PROVIDER.
SqlParser	Added support for NATIVE_COMPILATION and SCHEMABINDING options to CREATE TRIGGER T-SQL syntax.

Known issues 20.1

Expand table

Feature	Details	Workaround
Analysis Services	When you connect to Analysis Services with Microsoft Entra MFA, if you add a new role or open properties for a role, the message "the identity of the user being added to the role is not fetched properly" appears.	This error is benign and can be ignored. The error is addressed within the Azure infrastructure, and no updates to SSMS are required.

Feature	Details	Workaround
Analysis Services	After adding a new role, or when opening properties for an existing role, you can't use **Search by name or email address** to add a user.	A user can be added with the **Manual Entry** option.
Database Designer	Selecting the Design option for a view referencing a table using spatial data causes SSMS to crash.	Use T-SQL to make changes to the view.
Database Mirroring	If you launch the Database Mirroring Monitor from the mirrored node, the primary node isn't listed.	Register the mirrored node from Database Mirroring Monitoring, or use SSMS 18.12.1 to monitor from the mirrored node.
General SSMS	Import settings from SSMS 17 option not available.	Settings can be imported from SSMS 18.
Link feature for Azure SQL Managed Instance	After you remove an existing mirroring endpoint certificate on SQL Server, link creation through the wizard might fail due to unestablished trust between SQL Server and Azure SQL Managed Instance, even though all checks are successful.	Use PowerShell command `Get-AzSqlInstanceServerTrustCertificate` to check whether SQL Server mirroring endpoint certificate named `<SQL_Server_Instance_Name>` exists in the Azure SQL Managed Instance. If so, use PowerShell command `Remove-AzSqlInstanceServerTrustCertificate` to remove it before a new link creation attempt.
Linked servers	Creating a linked server to Azure SQL Database with SQL Server selected as Server type connects to the `master` database.	To create a linked server to Azure SQL Database, select **Other data source** for the **Server type**, and select **Microsoft OLE DB Provider for SQL Server** or **Microsoft OLE DB Driver for SQL Server** as the **Provider**. Enter the logical server name in the Data source field and the database name in the Catalog field.
PolyBase	PolyBase node isn't visible in Object Explorer when you connect to SQL Server 2022 (16.x).	Use SSMS 18.12.1.
Profiler	The Profiler menu isn't localized.	No current alternative.
Query Store Reports	Track Queries report does not update when alternate Metric and Execution options are selected within Configure.	Use SSMS 19.x.

Feature	Details	Workaround
Replication	If Azure SQL Managed Instance is the publisher and SSMS is running on a machine that isn't in the same virtual network as the publisher, you aren't able to insert a tracer token via Replication Monitor.	To insert tracer tokens, use Replication Monitor in SSMS on a machine that is in the same virtual network as the Azure SQL Managed Instance publisher.
SSIS	When creating or modifying an SSIS job step in a SQL Agent job, you receive the error "A connection was successfully established with the server, but then an error occurred during the login process. (provider: SSL Provider, error: 0 - The certificate chain was issued by an authority that is not trusted.) (Framework Microsoft SqlClient Data Provider)" regardless of whether Optional or Mandatory is selected for the Encryption property.	Use SSMS 19.3 to create or modify SSIS job steps.
Stretch Database	Removed Stretch Database Wizard.	Use T-SQL to configure Stretch Database or use SSMS 18.9.1 or earlier to use the Stretch Database Wizard.

20.0

Download SSMS 20.0

- Release number: 20.0
- Build number: 20.0.70.0
- Release date: March 19, 2024

Available languages:

- Chinese (Simplified)
- Chinese (Traditional)
- English (United States)
- French
- German
- Italian
- Japanese
- Korean

- Portuguese (Brazil)
- Russian
- Spanish

What's new in 20.0

Expand table

Feature	Details
Connection	The connection security properties **Encryption** and **Trust server certificate** now exist on the main sign in page in the Connection dialog for easier access. For more information, see Connect with SQL Server Management Studio.
Connection	A new property, **Host Name in Certificate**, used with the *Strict (SQL Server 2022 and Azure SQL)* and *Mandatory* **Encryption** options, now exists on the Login page of the Connection dialog.
Connection	Added icons to the Query Editor status bar to indicate the encryption method used for the connection.
Connection	Added Microsoft Entra ID authentication to **New Login** wizard.
Drivers	Updated Microsoft.Data.SqlClient version to 5.1.4, which includes support for Strict encryption and Transport Layer Security (TLS) 1.3.
Libraries	Updated Server Management Objects (SMO) version to 171.30.0
Libraries	Updated DacFx version to 162.1.x.
Options	Introduced a new option, **Trust server certificate for imported connections**, in **Tools** > **Options** > **SQL Server Object Explorer** > **Commands** under **Connection security**. For more information, see Options (SQL Server Object Explorer - Commands).

Bug fixes in 20.0

Expand table

Feature	Details
Always Encrypted	The **New Column Master Key** dialog supports Azure Key Vault using role permissions for authorization.
Connection	Resolved an issue with SSMS crashing when trying to connect to Azure Storage because the user didn't have access to any containers within the storage account.

Feature	Details
Connection	Fixed an issue where users couldn't change their password with **Trust server certificate** enabled.
Link feature for Azure SQL Managed Instance	Improved handling for importing and deleting certificates.
Link feature for Azure SQL Managed Instance	Addressed issues related to text and images in the Link wizards.
Security	Addressed vulnerability CVE-2023-2975 for the OpenSSL library.

Known issues 20.0

Expand table

Feature	Details	Workaround
Analysis Services	When you connect to Analysis Services with Microsoft Entra MFA, if you add a new role or open properties for a role, the message "the identity of the user being added to the role is not fetched properly" appears.	This error is benign and can be ignored. The error is addressed within the Azure infrastructure, and no updates to SSMS are required.
Analysis Services	After adding a new role, or when opening properties for an existing role, you can't use **Search by name or email address** to add a user.	A user can be added with the **Manual Entry** option.
Database Designer	Selecting the Design option for a view referencing a table using spatial data causes SSMS to crash.	Use T-SQL to make changes to the view.
Database Mirroring	If you launch the Database Mirroring Monitor from the mirrored node, the primary node isn't listed.	Register the mirrored node from Database Mirroring Monitoring, or use SSMS 18.12.1 to monitor from the mirrored node.
General SSMS	Import settings from SSMS 17 option not available.	Settings can be imported from SSMS 18.
Link feature for Azure SQL Managed Instance	After you remove an existing mirroring endpoint certificate on SQL Server, link creation through the wizard might fail due to	Use PowerShell command `Get-AzSqlInstanceServerTrustCertificate` to check whether SQL Server mirroring endpoint certificate named `<SQL_Server_Instance_Name>` exists in the

Feature	Details	Workaround
	unestablished trust between SQL Server and Azure SQL Managed Instance, even though all checks are successful.	Azure SQL Managed Instance. If so, use PowerShell command `Remove-AzSqlInstanceServerTrustCertificate` to remove it before a new link creation attempt.
Linked servers	Creating a linked server to Azure SQL Database with SQL Server selected as Server type connects to the `master` database.	To create a linked server to Azure SQL Database, select **Other data source** for the **Server type**, and select **Microsoft OLE DB Provider for SQL Server** or **Microsoft OLE DB Driver for SQL Server** as the **Provider**. Enter the logical server name in the Data source field and the database name in the Catalog field.
Maintenance Plans	Selecting the **Files and Filegroups** radio button in the Backup Database Task causes the dialog to close unexpectedly.	No current alternative.
PolyBase	PolyBase node isn't visible in Object Explorer when you connect to SQL Server 2022 (16.x).	Use SSMS 18.12.1.
Profiler	The Profiler menu isn't localized.	No current alternative.
Query Store Reports	Track Queries report does not update when alternate Metric and Execution options are selected within Configure.	Use SSMS 19.x.
Replication	If Azure SQL Managed Instance is the publisher and SSMS is running on a machine that isn't in the same virtual network as the publisher, you aren't able to insert a tracer token via Replication Monitor.	To insert tracer tokens, use Replication Monitor in SSMS on a machine that is in the same virtual network as the Azure SQL Managed Instance publisher.
SSIS	When creating or modifying an SSIS job step in a SQL Agent job, you receive the error "A connection was successfully established with the server, but then an error occurred during the login process. (provider: SSL Provider, error: 0 - The certificate chain was issued by an authority that is not trusted.) (Framework Microsoft SqlClient Data Provider)" regardless of whether	Use SSMS 19.3 to create or modify SSIS job steps.

Feature	Details	Workaround
	Optional or Mandatory is selected for the Encryption property.	
Stretch Database	Removed Stretch Database Wizard.	Use T-SQL to configure Stretch Database or use SSMS 18.9.1 or earlier to use the Stretch Database Wizard.

Known issues using Strict Encryption in 20.0

Expand table

Feature	Details	Workaround
Connection	When SQL Server is configured with **Force Strict Encryption**, selecting **Azure Data Studio** > **New Query** from the server or database menu generates the error, "A connection was successfully established with the server, but then an error occurred during the pre-login handshake. (provider: TCP provider, error: 0 - An existing connection was forcibly closed by the remote host.)"	Update the connection to use *Strict* instead of *Mandatory* for the **Encrypt** property in Azure Data Studio, and then connect.
Connection	Connecting to SQL Server with *Strict (SQL 2022 and Azure SQL)* selected for **Encryption** and a non-TCP/IP network protocol generates the error, "Cannot connect to SERVERNAME. A connection was successfully established with the server, but then an error occurred during the pre-login handshake. (provider: Shared Memory Provider, error: 15 - Function not supported) (Microsoft SQL Server, Error: 50) The request is not supported"	Change the **Network protocol** connection property to use *TCP/IP*, or enable the TCP/IP protocol for the SQL Server.
Database Tuning Advisor	When SQL Server is configured with **Force Strict Encryption**, connection to the server from the Database Tuning Advisor isn't supported.	No alternative.
Maintenance Plans	When you connect to a server with *Strict (SQL Server 2022 and Azure SQL)* encryption, modifying an existing maintenance plan generates the error "Failed to connect to SERVERNAME. (Microsoft.SqlServer.ConnectionInfo) A connection was successfully established with the server, but then an error occurred during the login process. (provider: SSL Provider, error: 0 - The target principal name is incorrect.)"	The problem doesn't occur when you connect with *Mandatory* or *Optional* encryption.
Profiler	When SQL Server is configured with **Force Strict Encryption**, connection to the server from Profiler isn't supported, and the error "Cannot connect to SERVERNAME. Class not registered (pfutil)" is generated.	Install MSOLEDBSQL version 19, available from Download Microsoft OLE DB

Feature	Details	Workaround
		Driver for SQL Server.
Profiler	When you connect to a server with *Strict (SQL Server 2022 and Azure SQL)* encryption and MSOLEDBSQL version 19 installed, traces can't be saved to, or loaded from, a database table.	No alternative.
PowerShell	When you connect to a server with *Strict (SQL Server 2022 and Azure SQL)* encryption, selecting **Start Powershell** from a node in Object Explorer generates the error "SQL Server PowerShell provider error: Could not connect to SERVERNAME. [Failed to connect to server SERVERNAME. --> A connection was successfully established with the server, but then an error occurred during the pre-login handshake. (provider: TCP Provider, error: 0 - An existing connection was forcibly closed by the remote host.) --> An existing connection was forcibly closed by the remote host]".	No current alternative.
SQL Server Logs	When SQL Server is configured with **Force Strict Encryption**, you can't view the SQL Server ERRORLOG files via Object Explorer, or executing `master.dbo.sp_enumerrorlogs` or `sys.xp_enumerrorlogs` via the Query Editor.	View the ERRORLOG files in the Log folder using File Explorer.

19.3

Download SSMS 19.3

- Release number: 19.3
- Build number: 19.3.4.0
- Release date: January 10, 2024

Available languages:

- Chinese (Simplified)
- Chinese (Traditional)
- English (United States)
- French
- German
- Italian
- Japanese
- Korean
- Portuguese (Brazil)
- Russian
- Spanish |

What's new in 19.3

Expand table

Feature	Details
Azure Data Studio installation integration	The installation of SSMS installs Azure Data Studio 1.47.1.

Bug fixes in 19.3

Expand table

Feature	Details
Accessibility	Addressed issue with screen reader announcing incomplete or incorrect information in the database properties dialog.
Always Encrypted	Fixed inability to change the **Enable Secure Enclaves** option when creating a new Azure SQL Database using a non-English installation of SSMS.
Availability Groups	Changed text color for primary server name in the Availability Group Dashboard, which caused the entry to appear empty.
Extended Events	Changed text color for data column after selecting **View Target Data** for the ring_buffer target.
Installer	Fixed issue where users might be prompted to update SQL Server Management Studio even if the current release is installed, see Bug in 19.2.56.2 update version detection.
Object Explorer	Resolved crash occurring when trying to close Object Explorer while the tree is still expanding.
Reports	Updated Server Dashboard report to correctly show the number of schedulers when more than 255 are available, see SSMS 19.2 - Reports - Server dashboard - Processors used by instance - wrong number.
Security	Update to Microsoft.Data.SqlClient 3.1.5 to address CVE-2024-0056.

Known issues 19.3

Expand table

Feature	Details	Workaround
Analysis Services	When you connect to Analysis Services with Microsoft Entra MFA, if you	This error is benign and can be ignored. It will be addressed within the Azure infrastructure soon, and no updates to SSMS are required.

Feature	Details	Workaround
	add a new role or open properties for a role, the message "the identity of the user being added to the role is not fetched properly" appears.	
Analysis Services	After adding a new role, or when opening properties for an existing role, you can't use **Search by name or email address** to add a user.	A user can be added with the **Manual Entry** option.
Database Designer	Selecting the Design option for a view that references a table using spatial data causes SSMS to crash.	Use T-SQL to make changes to the view.
Database Mirroring	When you launch the Database Mirroring Monitor from the mirrored node, the primary node isn't listed.	Use SSMS 18.12.1 if you need to monitor mirroring from the mirrored node.
General SSMS	Import settings from SSMS 17 option not available.	Settings can be imported from SSMS 18.
Link feature for Azure SQL Managed Instance	After you remove an existing mirroring endpoint certificate on SQL Server, link creation through the wizard might fail due to unestablished trust between SQL Server and Azure SQL Managed Instance, even though all checks are successful.	Use PowerShell command `Get-AzSqlInstanceServerTrustCertificate` to check whether SQL Server mirroring endpoint certificate named "<SQL_Server_Instance_Name>" exists in the Azure SQL Managed Instance. If so, use PowerShell command `Remove-AzSqlInstanceServerTrustCertificate` to remove it before a new link creation attempt.
Linked servers	Creating a linked server to Azure SQL Database with SQL Server selected as Server type connects to the `master` database.	To create a linked server to Azure SQL Database, select **Other data source** for the **Server type**, and select **Microsoft OLE DB Provider for SQL Server** or **Microsoft OLE DB Driver for SQL Server** as the **Provider**. Enter logical server name in the Data source field, and enter database name in the Catalog field.
PolyBase	PolyBase node isn't visible in Object Explorer when connecting to SQL 2022.	Use SSMS 18.12.1.

Feature	Details	Workaround
Profiler	The Profiler menu isn't localized.	No current alternative.
Replication	If Azure SQL Managed Instance is the publisher and SSMS is running on a machine, which isn't in the same virtual network as the publisher, you aren't able to insert a tracer token via Replication Monitor.	To insert tracer tokens, use Replication Monitor in SSMS on a machine that is in the same virtual network as the Azure SQL Managed Instance publisher.
Stretch DB	Removed Stretch DB Wizard.	Use T-SQL to configure Stretch DB or use SSMS 18.9.1 or earlier to use the Stretch DB Wizard.

18.12.1

Download SSMS 18.12.1

- Release number: 18.12.1
- Build number: 15.0.18424.0
- Release date: June 21, 2022

Available languages:

- Chinese (Simplified)
- Chinese (Traditional)
- English (United States)
- French
- German
- Italian
- Japanese
- Korean
- Portuguese (Brazil)
- Russian
- Spanish

What's new in 18.12.1

Expand table

New Item	Details
Azure Data Studio installation integration	Installation of SSMS installs Azure Data Studio 1.37.

Bug fixes in 18.12.1

Expand table

New Item	Details
Always Encrypted	Fixed issue with Column Master Key creation generating an exception when using Azure Key Vault as the key store.
Data Classification	Fixed issue with "Couldn't load file or assembly 'Microsoft.Information.Protection', Version=1.10.98.0" after upgrading to SSMS 18.10 or higher. See Latest SSMS 18.11.1 breaks the Data Classification. Get missing assembly error after updating.
SSMS General	Resolved error related to dacpac deployment using the Deploy Data-tier application option in Azure SQL DB with MFA.

17.9.1

Download SSMS 17.9.1

- Release number: 17.9.1
- Build number: 14.0.17289.0
- Release date: November 21, 2018

Available languages:

- Chinese (Simplified)
- Chinese (Traditional)
- English (United States)
- French
- German
- Italian
- Japanese
- Korean
- Portuguese (Brazil)
- Russian
- Spanish

What's new in 17.9.1

SQL Server utility is no longer available in versions 17.x and newer.

Bug fixes in 17.9.1

- Fixed an issue where users could experience their connection being closed and reopened with each query invocation when using "Azure Active Directory - Universal with MFA support" authentication with the SQL query editor. Side effects of the connection closing included global temporary tables being dropped unexpectedly and sometimes a new session ID (SPID) given to the connection.
- Fixed a long outstanding issue where a restore plan would fail to find a restore plan or generate an inefficient one under certain conditions.
- Fixed issue in the "Import Data-tier Application" wizard, which could result in an error when connected to an Azure SQL Database.

> **Note**
>
> Non-English localized releases of SSMS 17.x require the KB 2862966 security update package if installed on: Windows 8, Windows 7, Windows Server 2012, and Windows Server 2008 R2.

Uninstall and reinstall SSMS 17.x

If your SSMS installation is having problems, and a standard uninstall and reinstall doesn't resolve them, you can first try repairing the Visual Studio 2015 IsoShell. If repairing the Visual Studio 2015 IsoShell doesn't resolve the problem, the following steps have been found to fix many random issues:

1. Uninstall SSMS the same way you uninstall any application (using **Add or remove programs**).

2. Uninstall Visual Studio 2015 IsoShell **from an elevated cmd prompt**:

   Windows Command Prompt

   ```
   PUSHD "C:\ProgramData\Package
   Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\redist"
   vs_isoshell.exe /Uninstall /Force /PromptRestart
   ```

3. Uninstall Microsoft Visual C++ 2015 Redistributable the same way you uninstall any application. Uninstall both x86 and x64 if they're on your computer.

4. Reinstall Visual Studio 2015 IsoShell **from an elevated cmd prompt**:

Windows Command Prompt

```
PUSHD "C:\ProgramData\Package
Cache\FE948F0DAB52EB8CB5A740A77D8934B9E1A8E301\redist"
vs_isoshell.exe /PromptRestart
```

5. Reinstall SSMS.

6. Upgrade to the latest version of the Visual C++ 2015 Redistributable if you're not currently up to date.

16.5.3

Download SSMS 16.5.3

- Release number: 16.5.3
- Build number: 13.0.16106.4
- Release date: January 30, 2017

Available languages:

- Chinese (Simplified)
- Chinese (Traditional)
- English (United States)
- French
- German
- Italian
- Japanese
- Korean
- Portuguese (Brazil)
- Russian
- Spanish

Bug fixes in 16.5.3

- Fixed issue introduced in SSMS 16.5.2, which was causing the expansion of the 'Table' node when the table had more than one sparse column.

- Users can deploy SQL Server Integration Services (SSIS) packages containing OData Connection Manager, which connects to a Microsoft Dynamics AX/CRM Online resource to the SSIS catalog. For more information, For details, see OData Connection Manager.

- Configuring Always Encrypted on an existing table fails with errors on unrelated objects.

- Configuring Always Encrypted for an existing database with multiple schemas doesn't work.
- The Always Encrypted, Encrypted Column wizard fails due to the database containing views that reference system views.
- When you encrypt with Always Encrypted, errors from refreshing modules after encryption are incorrectly handled.
- *Open recent* menu doesn't show recently saved files.
- SSMS is slow when right-clicking an index for a table (over a remote (Internet) connection).
- Fixed an issue with the SQL Designer scrollbar.
- Context menu for tables momentarily stops responding.
- SSMS occasionally throws exceptions in Activity Monitor and crashes.
- SSMS crashes with the error: "The process was terminated due to an internal error in the .NET Runtime at IP 71AF8579 (71AE0000) with exit code 80131506."

More downloads

For a list of all SQL Server Management Studio downloads, search the Microsoft Download Center.

For the latest release of SQL Server Management Studio, see Download SQL Server Management Studio (SSMS).

Related content

- Download SQL Server Management Studio (SSMS)
- Download Azure Data Studio
- Azure Data Studio release notes

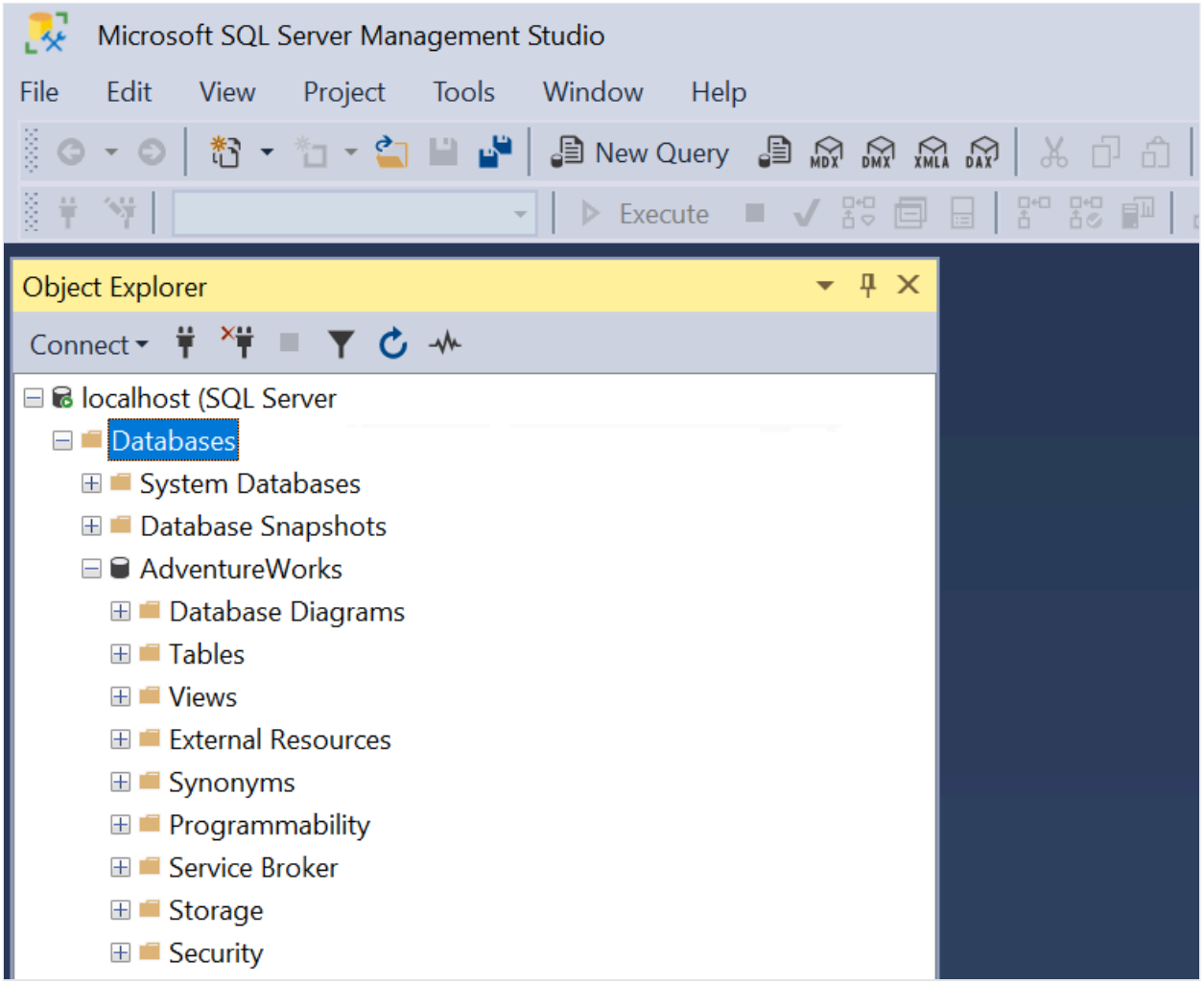
What is SQL Server Management Studio (SSMS)?

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

SQL Server Management Studio (SSMS) is an integrated environment for managing any SQL infrastructure. Use SSMS to access, configure, manage, administer, and develop all components of SQL Server, Azure SQL Database, Azure SQL Managed Instance, SQL Server on Azure VM, and Azure Synapse Analytics. SSMS provides a single comprehensive utility that combines a broad group of graphical tools with many rich script editors to provide access to SQL Server for developers and database administrators of all skill levels.

- **Download SQL Server Management Studio (SSMS)**
- **Download SQL Server Developer**
- **Download Visual Studio**



SQL Server Management Studio components

Expand table

Description	Component
Use **Object Explorer** to view and manage all objects in one or more instances of SQL Server.	Object Explorer
Use **Template Explorer** to build and manage files of boilerplate text that you use to speed the development of queries and scripts.	Template Explorer
Use the deprecated **Solution Explorer** to build projects to manage administration items such as scripts and queries.	Solution Explorer
Use the visual design tools included in SSMS to build queries, tables, and diagram databases.	Visual Database Tools
Use the SSMS language editors to interactively build and debug queries and scripts.	Query and Text Editors

SQL Server Management Studio for business intelligence

Use SSMS to access, configure, manage, and administer Analysis Services, Integration Services, and Reporting Services. Although all three business intelligence technologies rely on SSMS, the administrative tasks associated with each are slightly different.

> **Note**
>
> To create and modify Analysis Services, Integration Services solutions, and Reporting Services, use SQL Server Data Tools (SSDT), not SSMS. SQL Server Data Tools (SSDT) is a development environment that is based on Microsoft Visual Studio.

Manage Analysis Services solutions

SQL Server Management Studio (SSMS) enables you to manage Analysis Services objects, such as performing back-ups and processing objects.

SSMS provides an Analysis Services Script project in which you can develop and save scripts written in **Multidimensional Expressions (MDX)**, **Data Analysis Expressions**

(DAX), **Data Mining Extensions (DMX)**, and **XML for Analysis (XMLA)**.

These scripts are used to perform management tasks or recreate objects such as databases and instances on Analysis Services cubes. For example, you can develop an XMLA script in an Analysis Services Script project to create new objects directly on an existing instance. These projects can be saved as part of a solution and integrated with source code control.

> **Note**
>
> It is noted that while DAX was originally designed for tabular data models, it can also be used to query multidimensional models in SQL Server Analysis Services. SSMS can do DAX and MDX, but there are some considerations to keep in mind regarding the model you are working with and the type of queries you intend to run.

For more information about the Analysis Services Scripts Project in SSMS, see Analysis Services Scripts Project.

Manage Integration Services solutions

SQL Server Management Studio (SSMS) can be used to manage and monitor running SSIS packages. You can organize packages into folders, run, import, export, and upgrade Integration Services packages. However, since SSIS 2012, the storage of packages has changed. They're no longer stored in the server's `msdb` database of the default instance but are now managed through the SSIS Catalog database (`SSISDB`). This means that you can no longer manage packages in the same way as you did in previous versions of SSIS. You can still use SSMS to manage the SSIS Catalog database, but you must use the Integration Services Catalogs node in Object Explorer.

The latest version of SSMS provides an integrated environment for managing any SQL infrastructure. It also allows users to run SSIS packages stored in the SSIS Catalog from Object Explorer in SSMS.

The Import and Export Wizard within SSMS can be used to create SSIS packages, which is a good starting point for learning about SSIS. However, you must use SQL Server Data Tools (SSDT) to create and manage your packages for more complex packages.

Manage Reporting Services projects

SQL Server Management Studio (SSMS) enables Reporting Services features, administers the server and databases, and manages roles and jobs. You can manage shared schedules using the Shared Schedules folder and manage report server databases (`ReportServer`, `ReportServerTempDB`). When moving a report server database to a new SQL Server instance, you must create an RSExecRole in the `master` system database.

For more information on these tasks, you can refer to the articles on Reporting Services in SSMS, administering a Report Server database, and creating the **RSExecRole**:

- Reporting Services in SSMS
- Administer a Report Server database
- Create the RSExecRole

You also manage the server by enabling and configuring various features, setting server defaults, and managing roles and jobs.

For more information about these tasks, see the following articles:

- Set Report Server properties
- Create, delete, or modify a role
- Enabling and disabling client-side printing for Reporting Services

SSMS provides an integrated environment for managing any SQL infrastructure, including SSRS. In the web portal, you can enable features, set server defaults, manage running jobs, view custom reports, and create and manage shared schedules.

> ⓘ **Note**
>
> While SSMS offers various management capabilities for SSRS, it is not a replacement for Report Manager online or the Report Services Configuration Manager. It is recommended to stay updated with the latest versions and documentation to ensure effective management of SSRS within SSMS. If you have any specific questions or need further assistance, feel free to ask.

Non-English language versions

The block on mixed languages setup has been lifted. You can install SSMS in German on a French Windows machine. If the operating system language doesn't match the SSMS language, the user needs to change the language under Tools > Options > International Settings. Otherwise, SSMS shows the English UI.

For more information about different locales with previous versions, see Install non-English language versions of SSMS.

Support policy

Starting with SSMS 17.0, the SQL Tools team has adopted the Microsoft Modern Lifecycle Policy.

Read the original Modern Lifecycle Policy announcement. For more information, see Modern Policy FAQs.

For diagnostic data collection and feature usage information, see the SQL Server privacy supplement and diagnostic data collection.

Cross-platform tool

SSMS runs on Windows (AMD or Intel) only. If you need a tool that runs on platforms other than Windows, take a look at Azure Data Studio. Azure Data Studio is a cross-platform tool that runs on macOS, Linux, as well as Windows. For details, see Azure Data Studio.

Get help

- Ideas for SQL: Have suggestions for improving SQL Server?
- Microsoft Q & A (SQL Server)
- DBA Stack Exchange (tag sql-server): Ask SQL Server questions
- Stack Overflow (tag sql-server): Answers to SQL development questions
- Reddit: General discussion about SQL Server
- Microsoft SQL Server License Terms and Information
- Support options for business users
- Additional SQL Server help and feedback

Contribute to SQL documentation

Did you know that you can edit SQL content yourself? If you do so, not only do you help improve our documentation, but you also get credited as a contributor to the page.

For more information, see How to contribute to SQL Server documentation

Related content

- Install non-English language versions of SSMS
- Connect to and query a SQL Server instance
- Writing Transact-SQL Statements
- Azure Data Studio

Feedback

Was this page helpful? Yes No

Connect with SQL Server Management Studio

Article • 02/13/2025

This article describes how to connect to the SQL Server Database Engine with SQL Server Management Studio 20 and later versions, for the following products and services:

- SQL Server
- Azure SQL Database
- Azure SQL Managed Instance
- Azure Synapse Analytics

Encryption changes

SQL Server Management Studio (SSMS) 20 includes an important security improvement related to connection encryption, and it includes many changes.

- The **Encrypt connection** property has been renamed to **Encryption**.
- The value of *True* (**Encrypt connection** option checked, or enabled) now maps to *Mandatory* for **Encryption**.
- The value of *False* (**Encrypt connection** option unchecked, or disabled) now maps to *Optional* for **Encryption**.
- The **Encryption** property is now set to *Mandatory* by default.
- The **Encryption** property includes a new option, *Strict (SQL Server 2022 and Azure SQL)*, which isn't available in previous versions of SSMS.

In addition, the **Encryption** and **Trust server certificate** properties have moved to the **Login** page under **Connection Security**. A new option, **Host name in certificate**, has also been added. The best practice is to support a trusted, encrypted connection to the server.

To encrypt connections, the SQL Server Database Engine must be configured with a TLS certificate signed by a trusted root certificate authority.

For users connecting to Azure SQL Database and Azure SQL Managed Instance, existing connections in the most recently used (MRU) list should be updated to use *Strict (SQL Server 2022 and Azure SQL)* encryption. Any new connection to Azure SQL Database and Azure SQL Managed Instance should use *Strict (SQL Server 2022 and Azure SQL)*

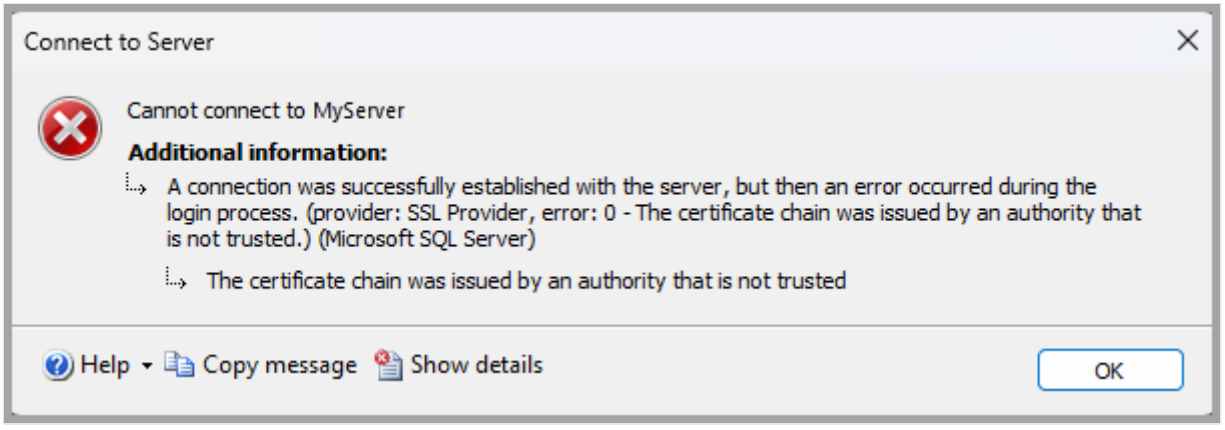
encryption. Azure SQL Database and Azure SQL Managed Instance support encrypted connections and are configured with trusted certificates.

With *Strict (SQL Server 2022 and Azure SQL)* encryption selected, the **Trust server certificate** option is unavailable.

With *Mandatory* encryption selected, and **Trust server certificate** enabled (checked), the **Host name in certificate** option is unavailable.

For users connecting to an on-premises SQL Server or SQL Server in a Virtual Machine, the Database Engine must be configured to support encrypted connections. For complete instructions, see Configure SQL Server Database Engine for encrypting connections.

If the Database Engine has either **Force Encryption** or **Force Strict Encryption** enabled, ensure that you have a certificate from a trusted certificate authority. For more information, see Certificate requirements for SQL Server. If you don't have a certificate from a trusted certificate authority and you try to connect, you see the error:



The text from the screenshot shows an error similar to the following example:

> **Cannot connect to MyServer**
>
> A connection was successfully established with the server, but then an error occurred during the login process. (provider: SSL Provider, error: 0 - > The certificate chain was issued by an authority that is not trusted.) (Microsoft SQL Server, Error: -2146893019) For help, click: `https://docs.microsoft.com/sql/relational-databases/errors-events/mssqlserver--2146893019-database-engine-error`
>
> The certificate chain was issued by an authority that is not trusted

For more information, see "The certificate received from the remote server was issued by an untrusted certificate authority" error when you connect to SQL Server.

New connections in SQL Server Management Studio 20 and later versions

For workstations with a new installation of SSMS, or when adding a new connection:

1. Select **Connect** in **Object Explorer**, or **File** > **New** > **Query with Current Connection**.
2. Within the **Connect to Server** dialog, set the connection details for your server.

Connection detail description

The following table describes the connection details.

Expand table

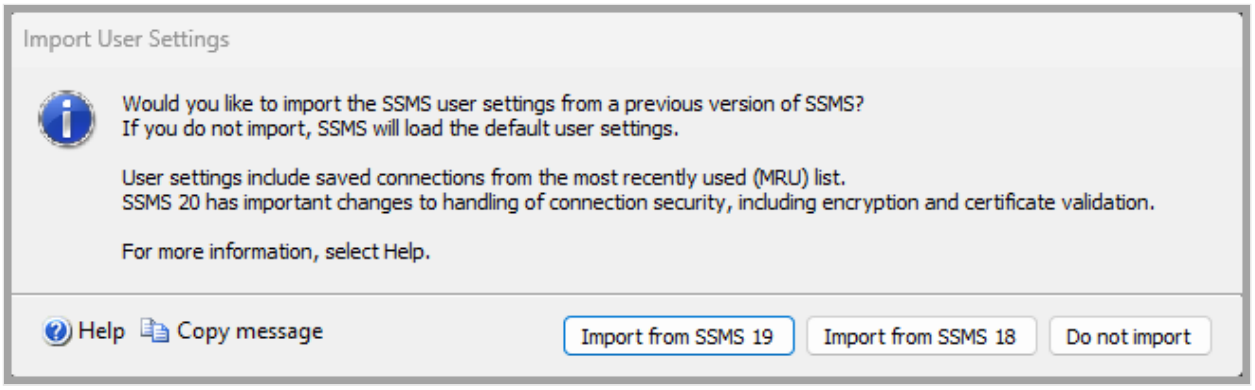
Setting	Description
Server name	Enter the server name here. For example, `MyServer` or `MyServer\MyInstance` or `localhost`.
Authentication	Select the type of authentication to use.
User name	Enter the user name with which to authenticate. The user name is automatically populated if *Windows Authentication* is selected for the **Authentication** type. When you use Microsoft Entra MFA, the user name can be left blank.
Password	Enter the password for the user (if available). The password field is unavailable for specific **Authentication** types.
***Encryption** [1]	Select the encryption level for the connection. The default value is *Mandatory*.
Trust server certificate	Check this option to bypass server certificate validation. The default value is *False* (unchecked), which promotes better security using trusted certificates.
Host name in the certificate	The value provided in this option is used to specify a different, but expected, CN or SAN in the server certificate for the server to which SSMS is connecting. This option can be left blank, so that certificate validation ensures that the Common Name (CN) or Subject Alternate Name (SAN) in the certificate matches the server name to which you're connecting. This parameter can be populated when the server name doesn't match the CN or SAN, for example, when using DNS aliases. For more information, see Encryption and certificate validation in Microsoft.Data.SqlClient.

[1] The value selected by the user represents the desired and minimal level of encryption. SSMS negotiates with the SQL engine (via the driver) to determine the encryption used,

and the connection can be established with a different (yet more secure) level of encryption. For example, if you select *Optional* for the **Encryption** option, and the server has **Force Encryption** enabled and a trusted certificate, then the connection is encrypted.

Connections after upgrading to SQL Server Management Studio 20

During the initial launch of SQL Server Management Studio 20, you're prompted to import settings from an earlier version of SSMS. If you select **Import from SSMS 19** or **Import from SSMS 18**, the list of most recently used connections is imported from the selected version of SSMS.



For workstations that imported settings from SSMS 18 or SSMS 19:

1. Select **Connect** in **Object Explorer**, or **File** > **New** > **Query with Current Connection**.
2. Within the **Connect to Server** dialog, select the server to which you want to connect.
3. Review the options selected for **Encryption** and **Trust server certificate** for the existing connection. In some scenarios, it might be necessary to configure a signed certificate or change the value for one or both properties.
4. If you previously connected with **Encryption** set to *Mandatory*, and the SQL Server doesn't have encrypted connections enabled or doesn't have a signed certificate, you're prompted to enable the **Trust server certificate** option:

1. If you select yes, **Trust Server Certificate** is enabled for the connection. You can also enable **Trust Server Certificate** for all connections imported from older versions of SSMS.
2. If you select no, **Trust Server Certificate** isn't enabled, and you aren't able to connect. Review the configuration to verify that a valid server certificate is installed.

For more information, see Configure SQL Server Database Engine for encrypting connections.

Remarks

These changes are the result of updates at the driver level in Microsoft.Data.SqlClient. Recent releases of Microsoft.Data.SqlClient offer increased security in the connection options. For more information, see Introduction to Microsoft.Data.SqlClient namespace.

Related content

- Quickstart: Connect and query a SQL Server instance using SQL Server Management Studio (SSMS)
- Quickstart: Connect and query a SQL Server instance on an Azure Virtual Machine using SQL Server Management Studio (SSMS)
- Quickstart: Connect and query an Azure SQL Database or an Azure SQL Managed Instance using SQL Server Management Studio (SSMS)
- Quickstart: Connect and query a dedicated SQL pool (formerly SQL DW) in Azure Synapse Analytics with SQL Server Management Studio (SSMS)

Feedback

Was this page helpful? Yes No

Quickstart: Connect and query a SQL Server instance using SQL Server Management Studio (SSMS)

Article • 02/13/2025

Applies to: SQL Server

Get started using SQL Server Management Studio (SSMS) to connect to your SQL Server instance and run some Transact-SQL (T-SQL) commands.

> **Note**
>
> While Microsoft Entra ID is the new name for Azure Active Directory (Azure AD), to prevent disrupting existing environments, Azure AD still remains in some hardcoded elements such as UI fields, connection providers, error codes, and cmdlets. In this article, the two names are interchangeable.

The article demonstrates how to follow the below steps:

- ✓ Connect to a SQL Server instance
- ✓ Create a database
- ✓ Create a table in your new database
- ✓ Insert rows into your new table
- ✓ Query the new table and view the results
- ✓ Use the query window table to verify your connection properties

This article covers connecting and querying an instance of SQL Server. For Azure SQL, see Connect and query Azure SQL Database & SQL Managed Instance.

To use Azure Data Studio, see connect and query SQL Server, Azure SQL Database, and Azure Synapse Analytics.

To learn more about SQL Server Management Studio, see Tips and tricks for using SQL Server Management Studio (SSMS).

Prerequisites

To complete this quickstart, you need the following prerequisites:

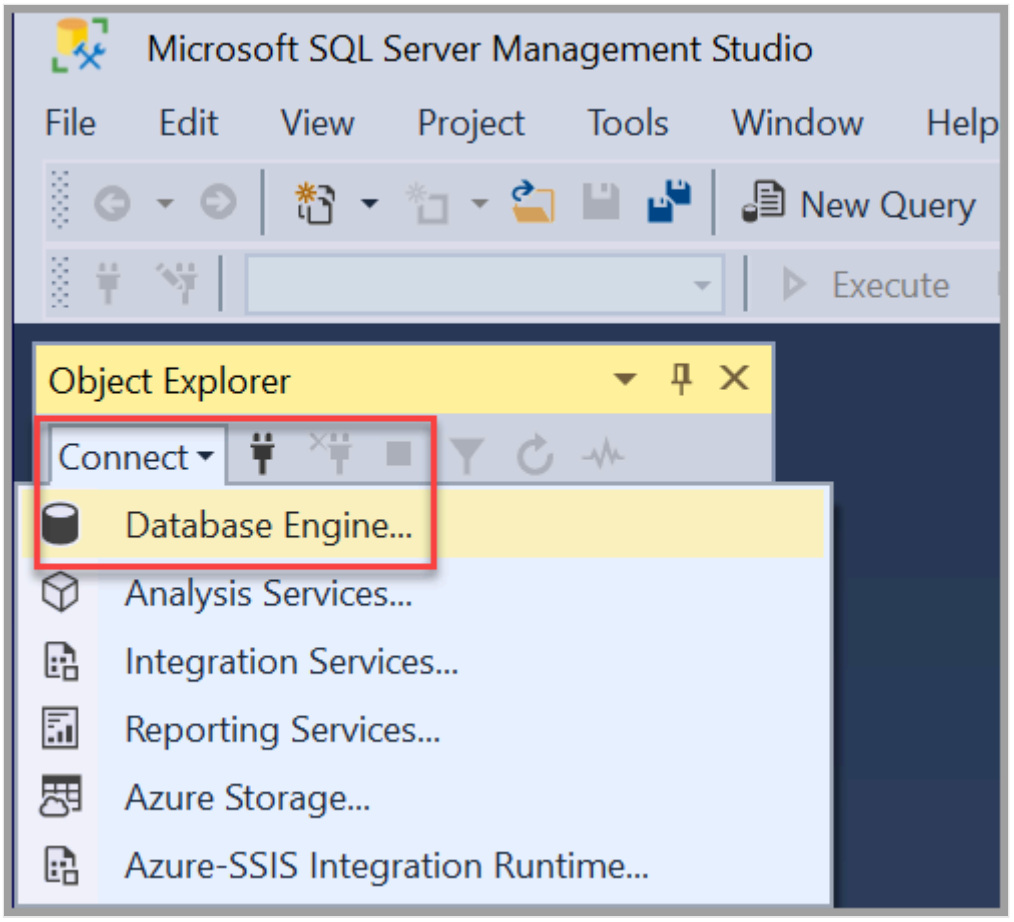
- Install Download SQL Server Management Studio (SSMS).

- Install SQL Server from the Installation Wizard (Setup) and configure a SQL Server instance.

Connect to a SQL Server instance

To connect to your SQL Server instance, follow these steps:

1. Start SQL Server Management Studio. The first time you run SSMS, the **Connect to Server** window opens. If it doesn't open, you can open it manually by selecting **Object Explorer** > **Connect** > **Database Engine**.



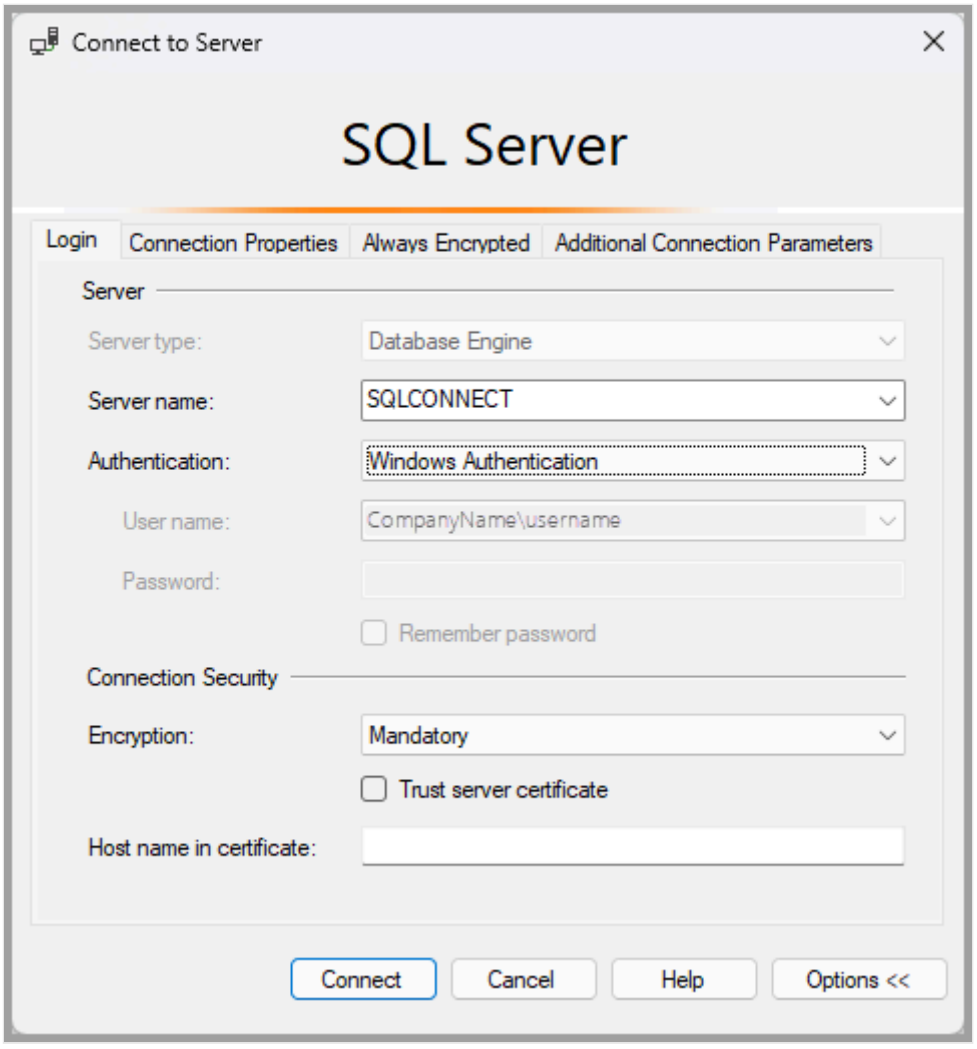
2. The **Connect to Server** dialog box appears. Enter the following information:

Expand table

Setting	Suggested values	Description
Server type	Database Engine	For **Server type**, select **Database Engine** (usually the default option).
Server name	The fully qualified server name	For **Server name**, enter the name of your SQL Server (you can also use *localhost* as the server name if you're connecting locally). If you're NOT using the

Setting	Suggested values	Description
		default instance - ***MSSQLSERVER*** - you must enter in the server name and the instance name. If you're unsure how to determine your SQL Server instance name, see Additional tips and tricks for using SSMS.
Authentication	Windows Authentication SQL Server Authentication Microsoft Entra authentication	Windows Authentication is set as default. You can also use **SQL Server Authentication** to connect. However, if you select **SQL Server Authentication**, a username and password are required. **Microsoft Entra authentication** is available for SQL Server 2022 (16.x) and later versions. For step-by-step configuration instructions, see Tutorial: Set up Microsoft Entra authentication for SQL Server For more information about authentication types, see Connect to the server (database engine).
Login	Server account user ID	The user ID from the server account used to sign in to the server. A login is required when using **SQL Server Authentication**.
Password	Server account password	The password from the server account used to sign in to the server. A password is required when using **SQL Server Authentication**.
Encryption [1]	Encryption method	Select the encryption level for the connection. The default value is *Mandatory*.
Trust server certificate	Trust Server Certificate	Check this option to bypass server certificate validation. The default value is *False* (unchecked), which promotes better security using trusted certificates.
Host Name in Certificate	Host name of the server	The value provided in this option is used to specify a different, but expected, CN or SAN in the server certificate.

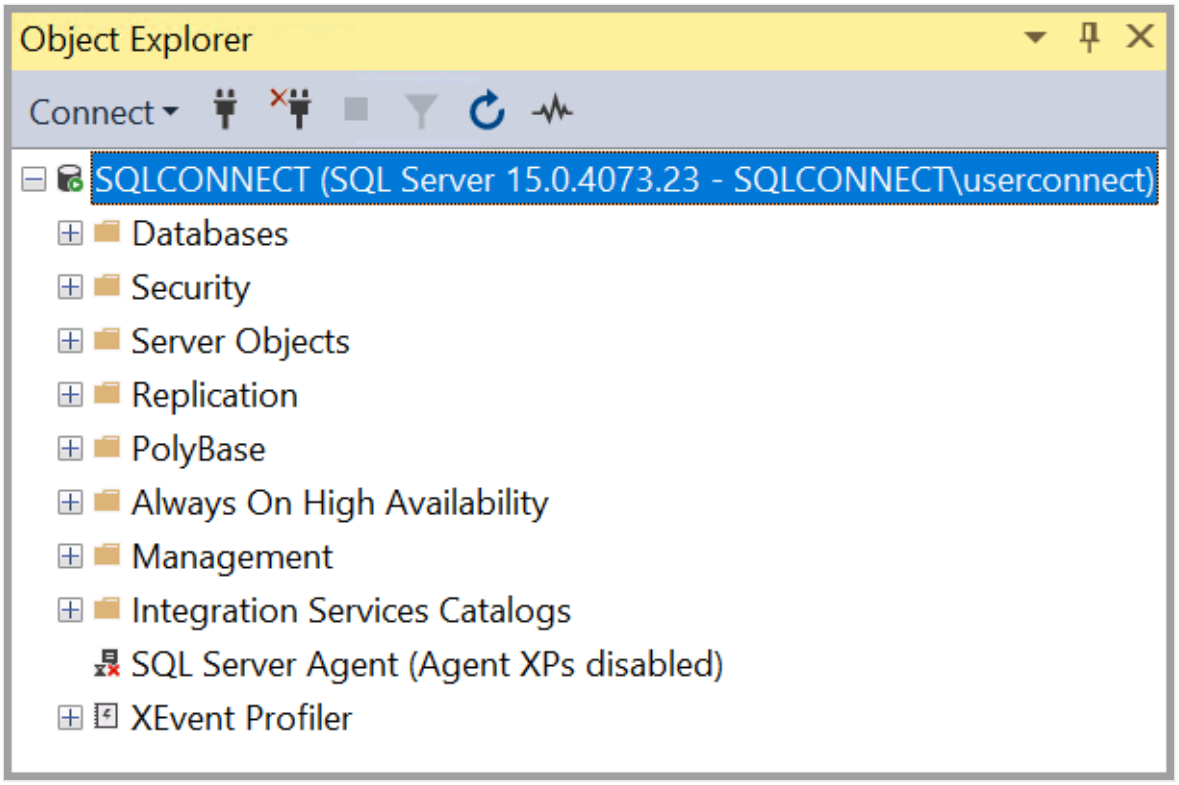
[1] The default value is *Mandatory* in SSMS 20. *Strict (SQL Server 2022 and Azure SQL)* encryption should be used for Azure SQL Database and Azure SQL Managed Instance. *Strict (SQL Server 2022 and Azure SQL)* encryption can be used for SQL Server when the instance has **Force Strict Encryption** enabled.



3. After you complete all the fields, select **Connect**.

   You can also modify extra connection options by selecting **Options**. Examples of connection options are the database you're connecting to, the connection timeout value, and the network protocol. This article uses the default values for all the fields.

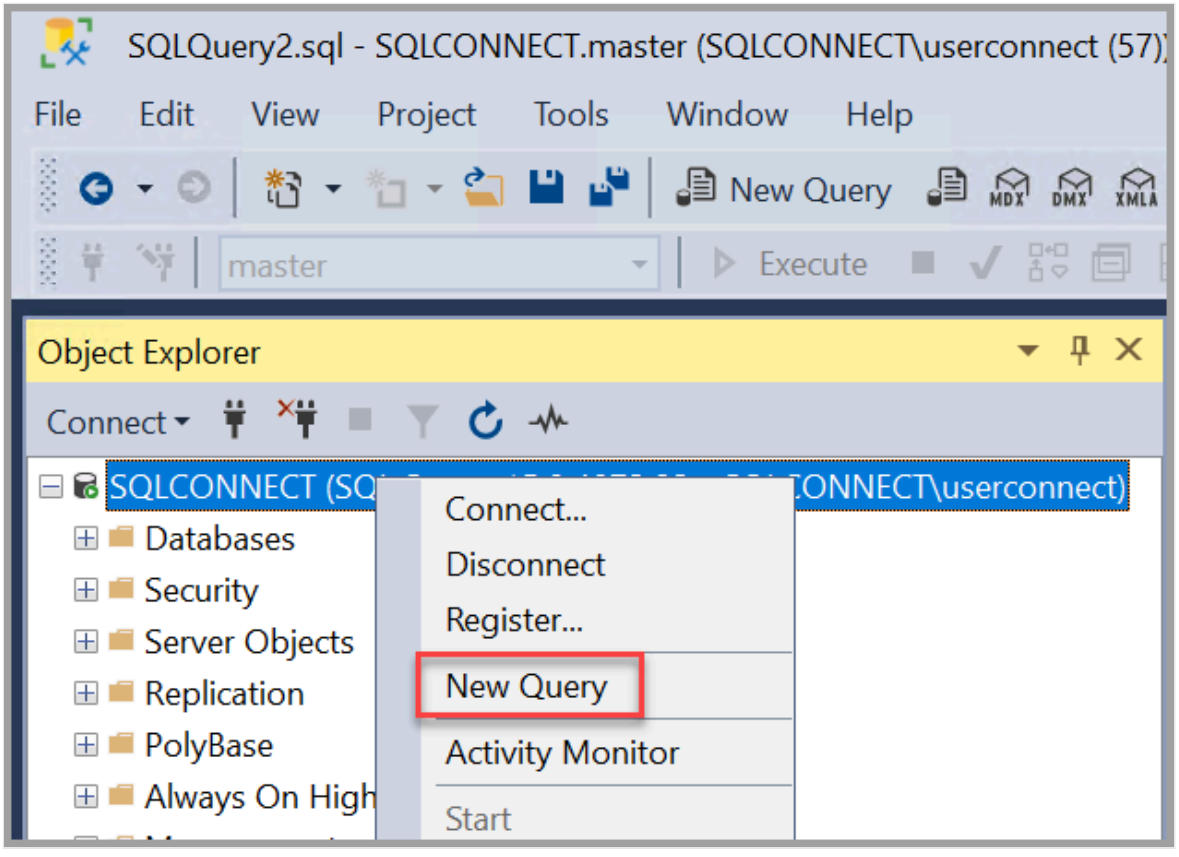
4. To verify that your SQL Server connection succeeded, expand and explore the objects within **Object Explorer** where the server name, the SQL Server version, and the username are displayed. These objects are different depending on the server type.



Create a database

Now let's create a database named TutorialDB by following the below steps:

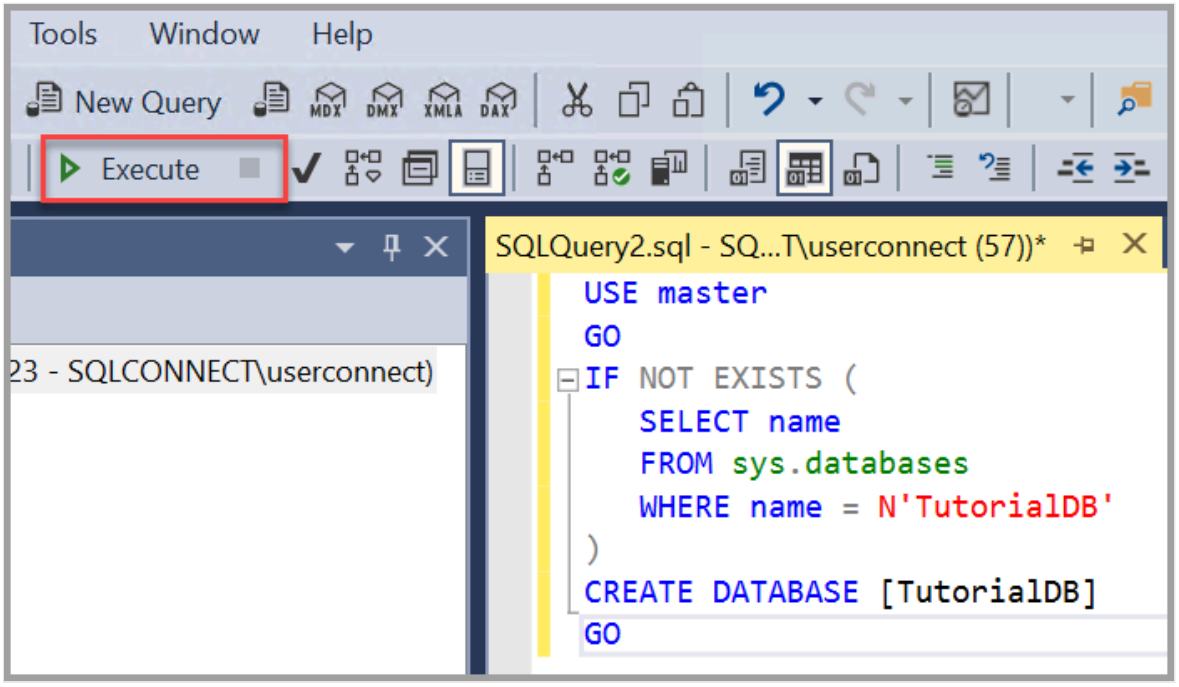
1. Right-click your server instance in Object Explorer, and then select **New Query**:



2. Paste the following T-SQL code snippet into the query window:

```SQL
USE master
GO
IF NOT EXISTS (
   SELECT name
   FROM sys.databases
   WHERE name = N'TutorialDB'
)
CREATE DATABASE [TutorialDB]
GO
```

3. Execute the query by selecting **Execute** or selecting F5 on your keyboard.

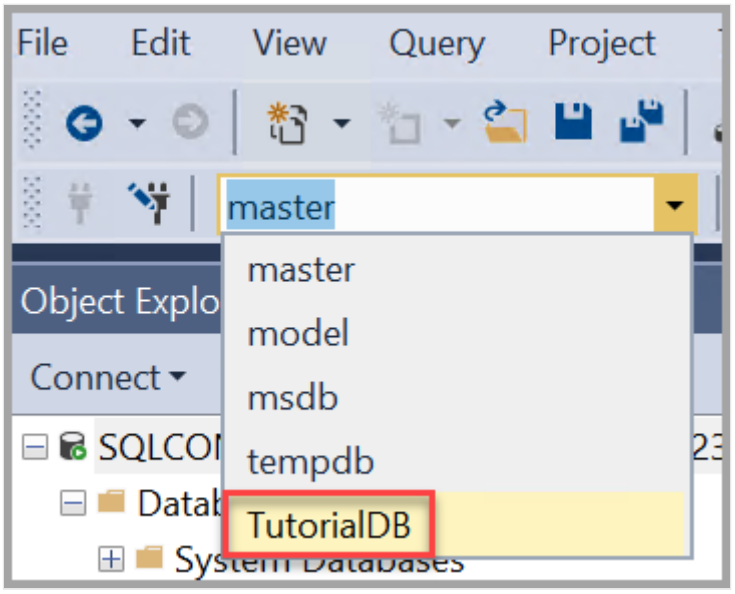


After the query is complete, the new TutorialDB database appears in the list of databases in Object Explorer. If it isn't displayed, right-click the **Databases** node, and then select **Refresh**.

Create a table

In this section, you create a table in the newly created TutorialDB database. Because the query editor is still in the context of the `master` database, switch the connection context to the *TutorialDB* database by doing the following steps:

1. In the database dropdown list, select the database that you want, as shown here:

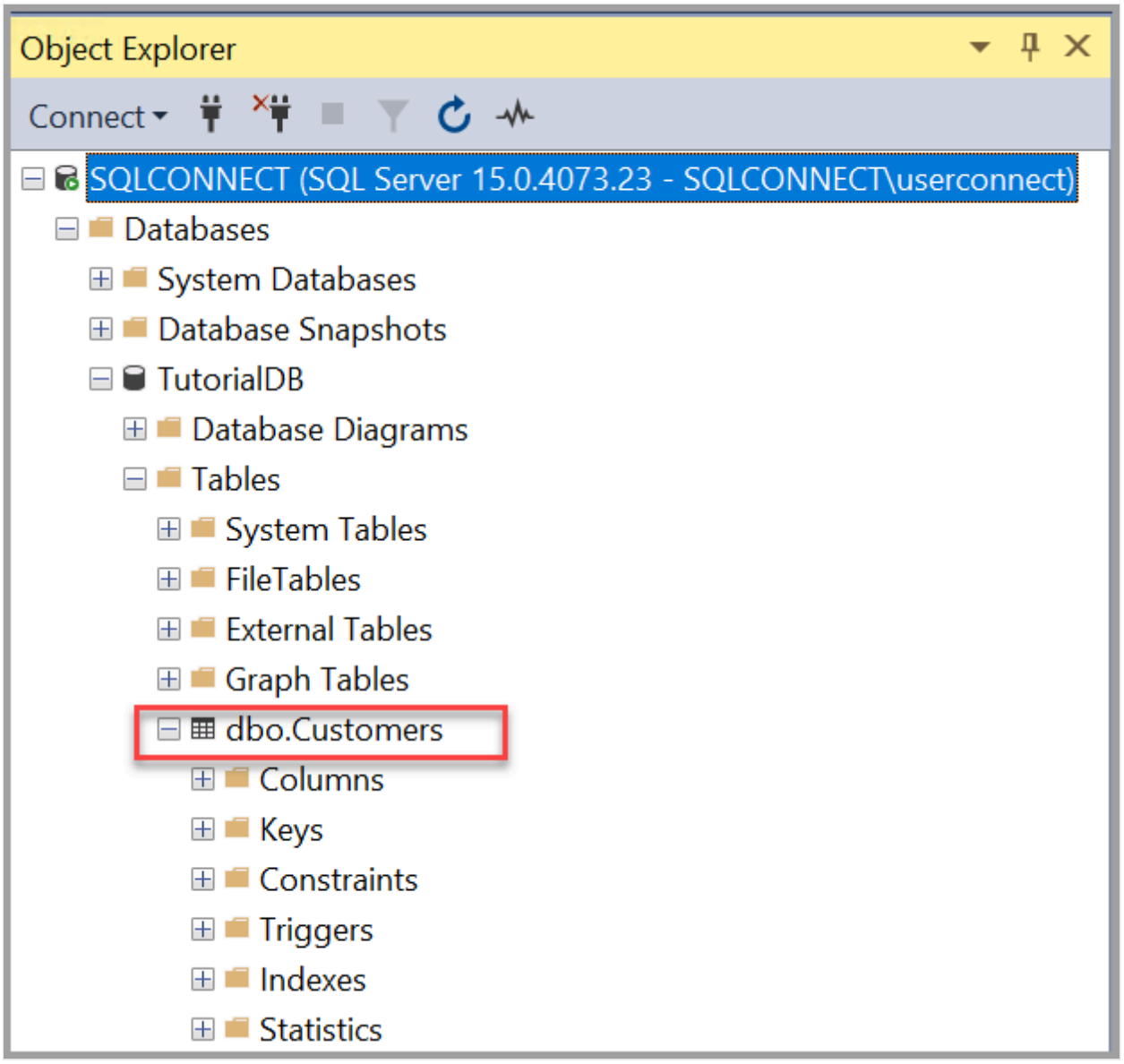


2. Paste the following T-SQL code snippet into the query window:

```SQL
USE [TutorialDB]

-- Create a new table called 'Customers' in schema 'dbo'
-- Drop the table if it already exists
IF OBJECT_ID('dbo.Customers', 'U') IS NOT NULL
    DROP TABLE dbo.Customers
GO

-- Create the table in the specified schema
CREATE TABLE dbo.Customers (
    CustomerId INT NOT NULL PRIMARY KEY, -- primary key column
    Name NVARCHAR(50) NOT NULL,
    Location NVARCHAR(50) NOT NULL,
    Email NVARCHAR(50) NOT NULL
);
GO
```

3. Execute the query by selecting **Execute** or selecting F5 on your keyboard.

After the query is complete, the new Customers table is displayed in the list of tables in Object Explorer. If the table isn't displayed, right-click the **TutorialDB** > **Tables** node in Object Explorer, and then select **Refresh**.



Insert rows

Now let's insert some rows into the Customers table that you created. Paste the following T-SQL code snippet into the query window, and then select **Execute**:

SQL

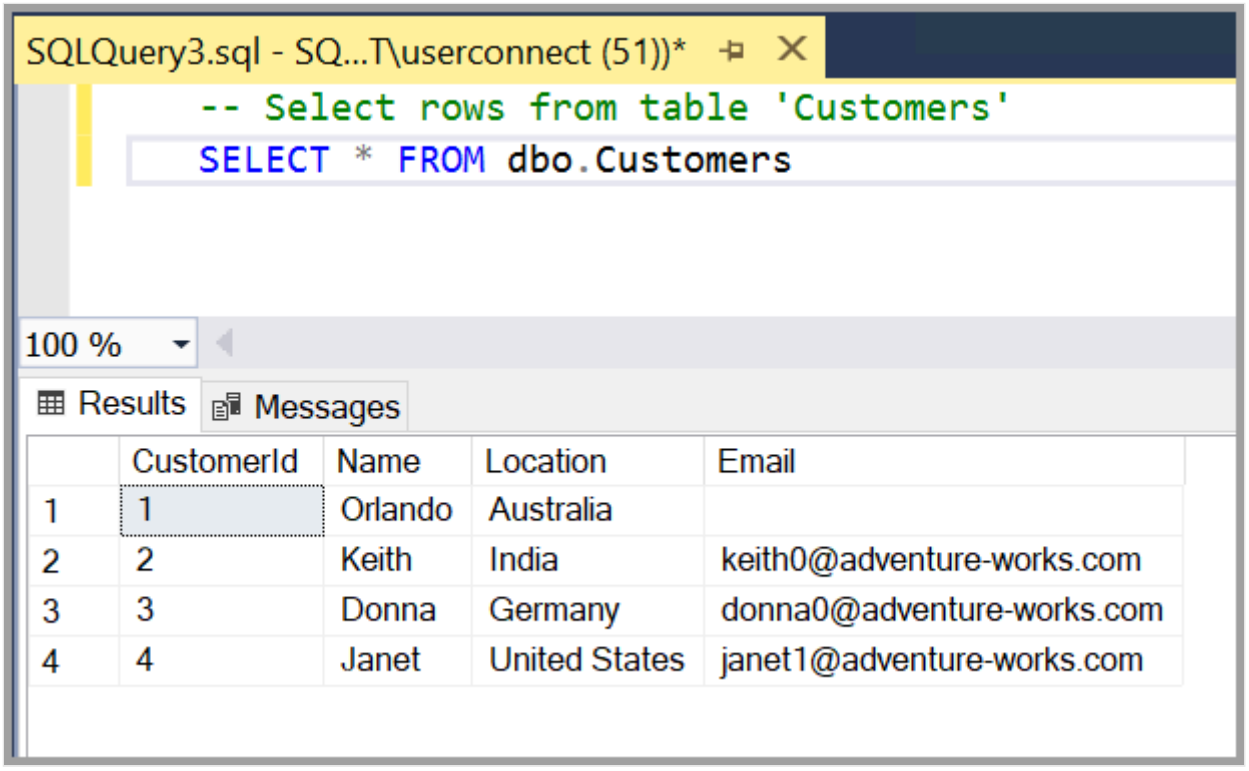
```
-- Insert rows into table 'Customers'
INSERT INTO dbo.Customers
   ([CustomerId],[Name],[Location],[Email])
VALUES
   ( 1, N'Orlando', N'Australia', N''),
   ( 2, N'Keith', N'India', N'keith0@adventure-works.com'),
   ( 3, N'Donna', N'Germany', N'donna0@adventure-works.com'),
   ( 4, N'Janet', N'United States', N'janet1@adventure-works.com')
GO
```

Query the table and view the results

The results of a query are visible below the query text window. To query the Customers table and view the rows that were inserted, paste the following T-SQL code snippet into the query window, and then select **Execute**:

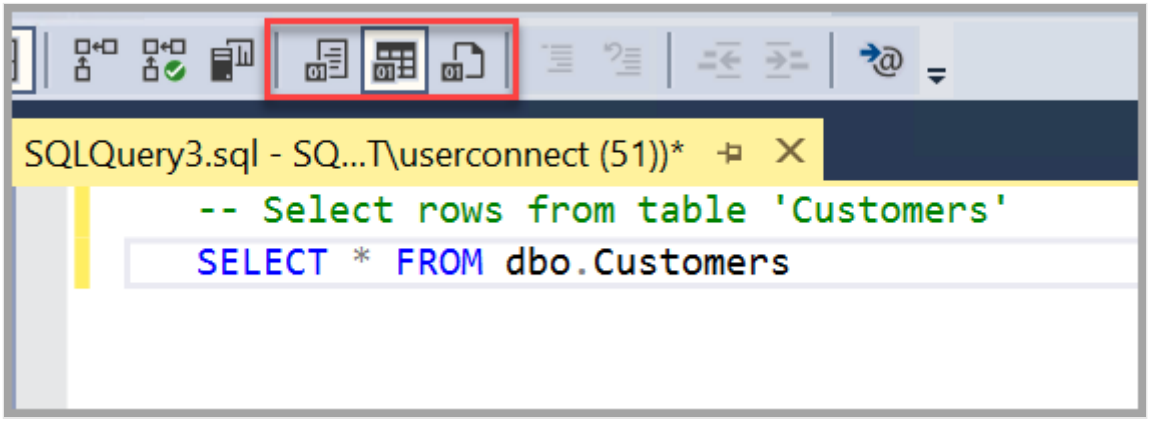
```SQL
-- Select rows from table 'Customers'
SELECT * FROM dbo.Customers;
```

The query results are displayed under the area where the text was entered.



	CustomerId	Name	Location	Email
1	1	Orlando	Australia	
2	2	Keith	India	keith0@adventure-works.com
3	3	Donna	Germany	donna0@adventure-works.com
4	4	Janet	United States	janet1@adventure-works.com

You can also modify the way results are presented by selecting one of the following options:



- The first button displays the results in **Text View**, as shown in the image in the next section.
- The middle button displays the results in **Grid View**, which is the default option.
- The third button lets you save the results to a file whose extension is .rpt by default.

Troubleshoot connectivity issues

To review troubleshooting techniques to use when you can't connect to an instance of your SQL Server Database Engine on a single server, visit Troubleshoot connecting to the SQL Server Database Engine.

Related content

- SQL Server Management Studio (SSMS) Query Editor
- Script objects in SQL Server Management Studio
- Use templates in SQL Server Management Studio
- SQL Server Management Studio components and configuration
- Tips and tricks for using SQL Server Management Studio (SSMS)

Feedback

Was this page helpful? Yes No

Quickstart: Connect and query a SQL Server instance on an Azure Virtual Machine using SQL Server Management Studio (SSMS)

Article • 02/13/2025

Applies to: SQL Server

Get started using SQL Server Management Studio (SSMS) to connect to your SQL Server instance on an Azure Virtual Machine and run some Transact-SQL (T-SQL) commands.

> **Note**
>
> While Microsoft Entra ID is the new name for Azure Active Directory (Azure AD), to prevent disrupting existing environments, Azure AD still remains in some hardcoded elements such as UI fields, connection providers, error codes, and cmdlets. In this article, the two names are interchangeable.

- ✓ Connect to a SQL Server instance
- ✓ Create a database
- ✓ Create a table in your new database
- ✓ Insert rows into your new table
- ✓ Query the new table and view the results
- ✓ Use the query window table to verify your connection properties

Prerequisites

To complete this article, you need SQL Server Management Studio and access to a data source.

- Install SQL Server Management Studio (SSMS)
- SQL Server on an Azure VM

Connect to SQL Virtual Machines

The following steps show how to create an optional DNS label for your Azure VM and then connect with SQL Server Management Studio.

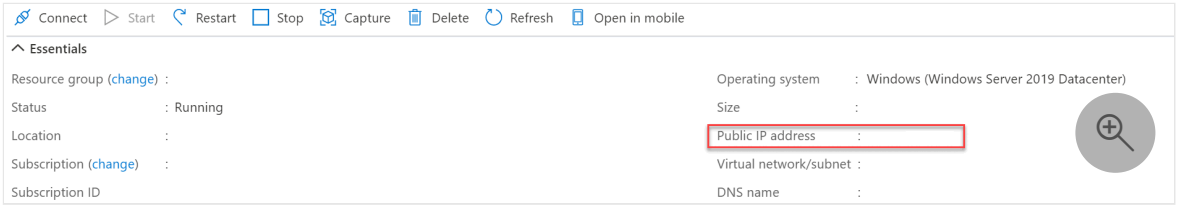
Configure a DNS Label for the public IP address

To connect to the SQL Server Database Engine from the Internet, consider creating a DNS Label for your public IP address. You can join by IP address, but the DNS Label creates an A record that is easier to identify and abstracts the underlying public IP address.

> **Note**
>
> DNS Labels aren't required if you plan to only connect to the SQL Server instance within the same Virtual Network or only locally.

1. Create a DNS Label by selecting **Virtual machines** in the portal. Select your SQL Server VM to bring up its properties.

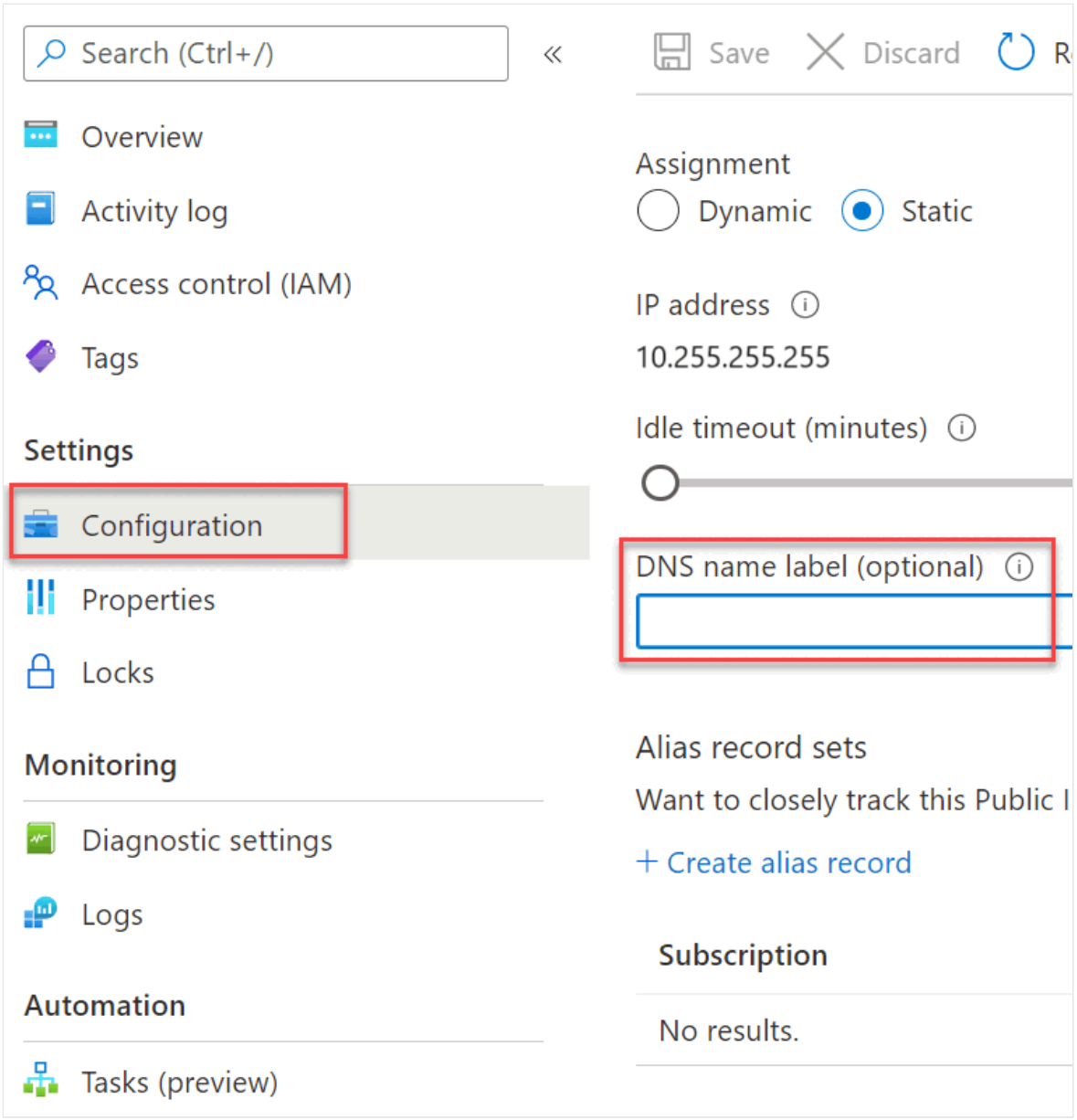
2. In the virtual machine overview, select your **Public IP address**.



3. In the properties for your Public IP address, expand **Configuration**.

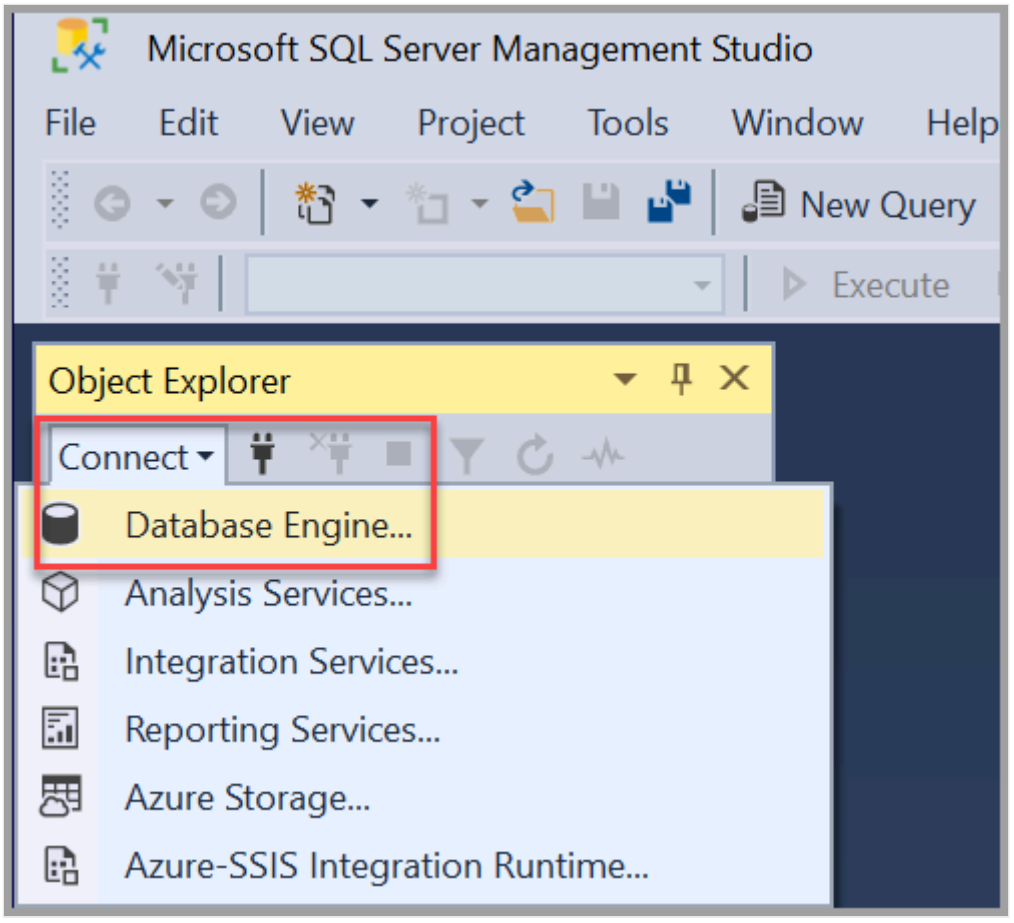
4. Enter a DNS Label name. This name is an A record that can be used to directly connect to your SQL Server VM by name instead of by IP address.

5. Select the **Save** button.



Connect

1. Start SQL Server Management Studio. The first time you run SSMS, the **Connect to Server** window opens. If it doesn't open, you can open it manually by selecting **Object Explorer** > **Connect** > **Database Engine**.



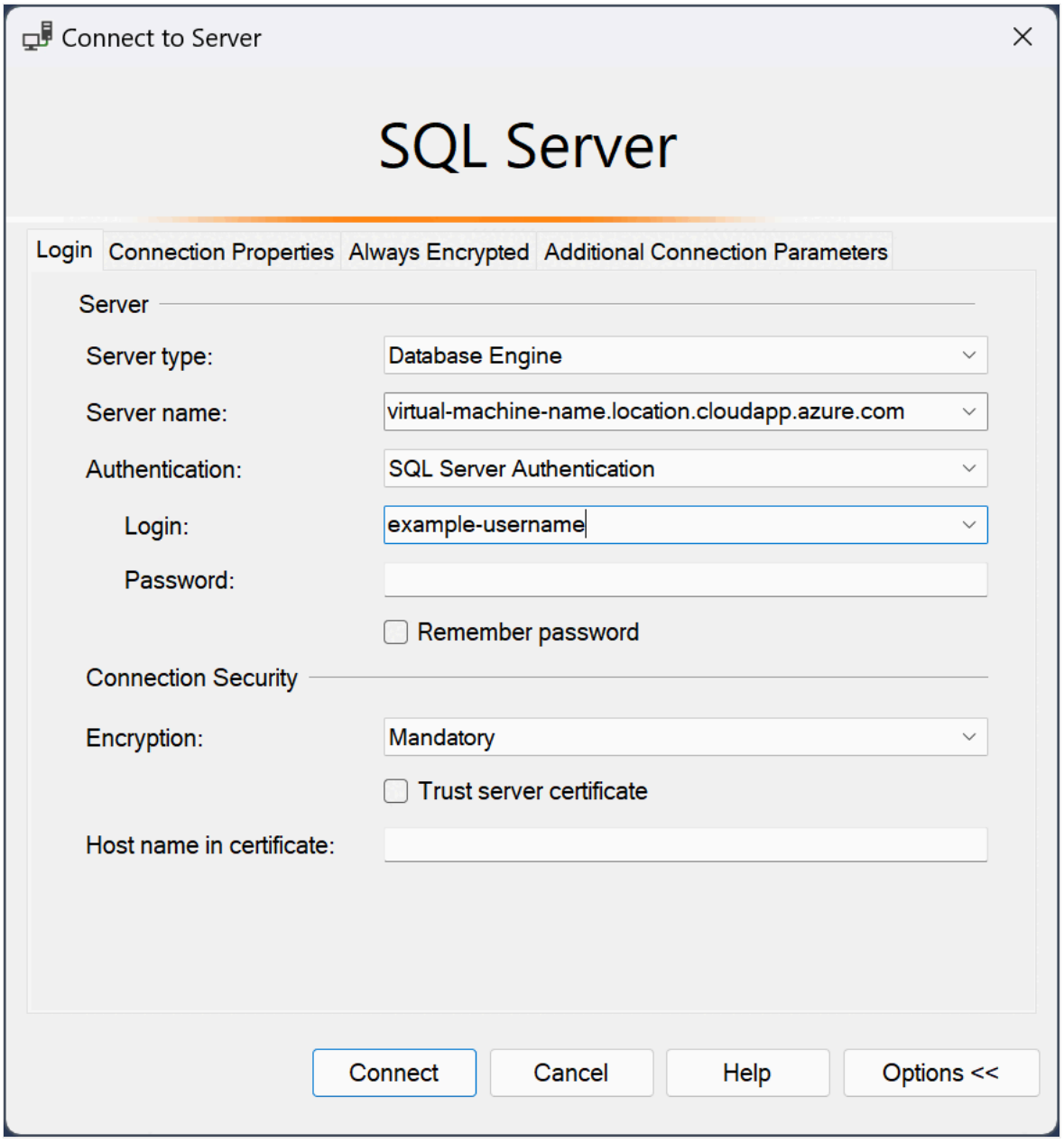
2. The **Connect to Server** dialog box appears. Enter the following information:

Expand table

Setting	Suggested values	Description
Server type	Database engine	For **Server type**, select **Database Engine** (usually the default option).
Server name	The fully qualified server name	For **Server name**, enter the name of your SQL Server VM. You can also use the SQL Server VM IP address to connect.
Authentication	Microsoft Entra - Managed Identity	Use **managed identity authentication** to connect to SQL Server VM. Using managed identity authentication eliminates the need for developer-managed credentials. Use SQL Server Authentication only when necessary. **Windows Authentication** isn't supported for SQL Server VM. For more information, see Azure SQL authentication.
Login	Server account user ID	The user ID from the server account used to create the server. A sign in is required when using **SQL**

Setting	Suggested values	Description
		Server Authentication.
Password	Server account password	The password from the server account used to create the server. A password is required when using **SQL Server Authentication**.
Encryption [1]	Encryption method	Select the encryption level for the connection. The default value is *Mandatory*.
Trust server certificate	Trust Server Certificate	Check this option to bypass server certificate validation. The default value is *False* (unchecked), which promotes better security using trusted certificates.
Host Name in Certificate	Host name of the server	The value provided in this option is used to specify a different, but expected, CN or SAN in the server certificate.

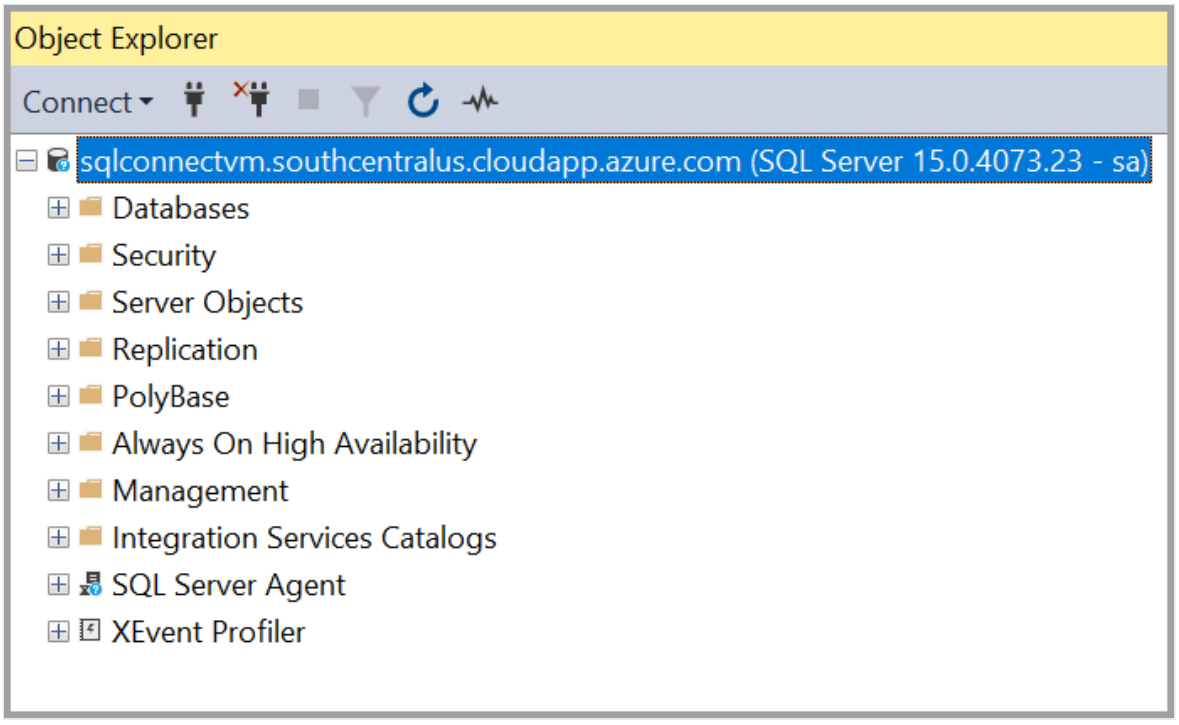
[1] The default value is *Mandatory* in SSMS 20. *Strict (SQL Server 2022 and Azure SQL)* encryption should be used for Azure SQL Database and Azure SQL Managed Instance. *Strict (SQL Server 2022 and Azure SQL)* encryption can be used for SQL Server when the instance has **Force Strict Encryption** enabled.



3. After you complete all the fields, select **Connect**.

   You can also modify additional connection options by selecting **Options**. Examples of connection options are the database you're connecting to, the connection timeout value, and the network protocol. This article uses the default values for all the options.

4. To verify that your SQL Server on Azure VM succeeded, expand and explore the objects within **Object Explorer** where the server name, the SQL Server version, and the username are displayed. These objects are different depending on the server type.



Troubleshoot connectivity issues

Although the portal provides options to configure connectivity automatically, knowing how to manually configure connectivity is useful. Understanding the requirements can also aid troubleshooting.

The following table lists the requirements to connect to SQL Server on Azure VM.

Expand table

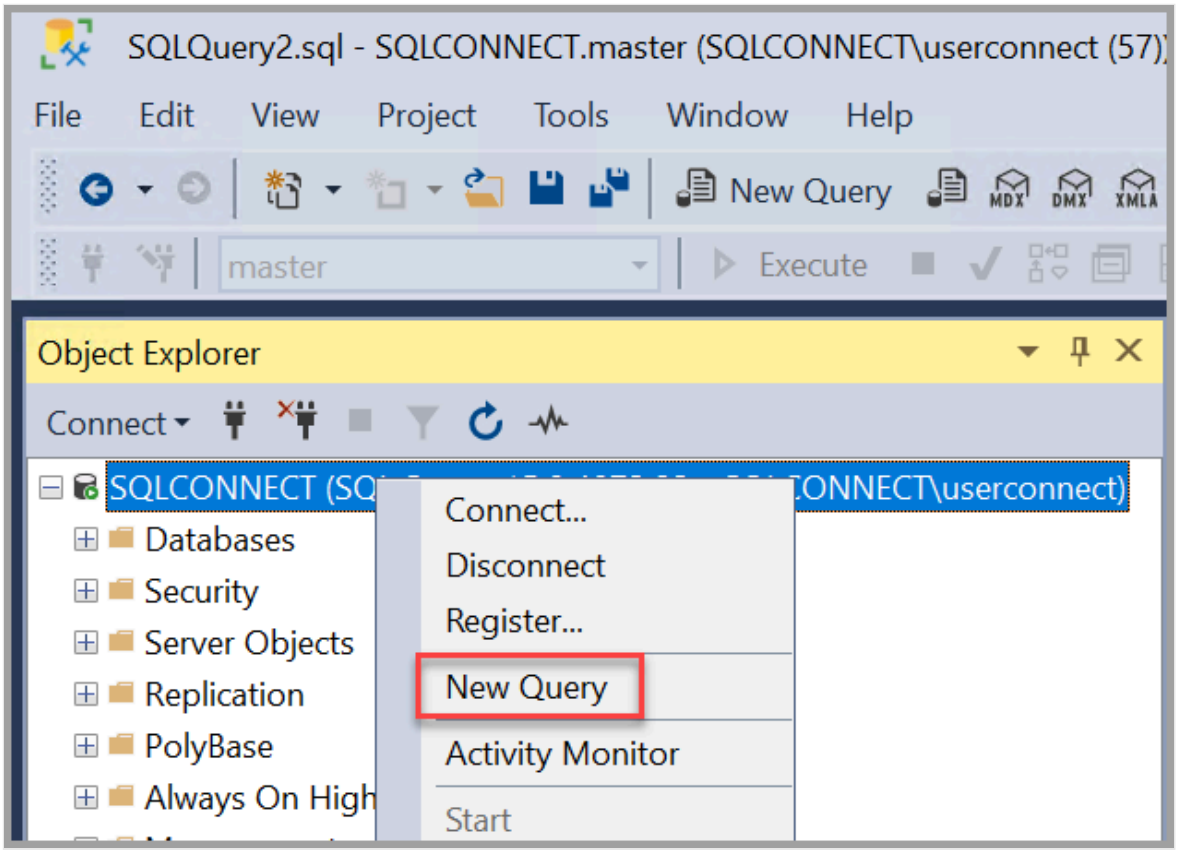
Requirement	Description
Enable SQL Server authentication mode	SQL Server authentication is needed to connect to the VM remotely unless you configured Active Directory on a virtual network.
Create a Login	If you're using SQL authentication, you need a SQL login with a user name and password that also has permissions to your target database.
Enable TCP/IP protocol	SQL Server must allow connections over TCP.
Configure a Windows Firewall for Database Engine Access	The firewall on the VM must allow inbound traffic on the SQL Server port (default 1433).
Create a network security group rule for TCP 1433	Allow the VM to receive traffic on the SQL Server port (default 1433) if you want to connect over the internet. This isn't required for local and virtual-network-only connections. This step is only required in the Azure portal.

> **Tip**
>
> The steps in the preceding table are done for you when you configure connectivity in the portal. Use these steps only to confirm your configuration or to set up connectivity manually for SQL Server.

Create a database

Create a database named `TutorialDB` by following the below steps:

1. Right-click your server instance in Object Explorer, and then select **New Query**:

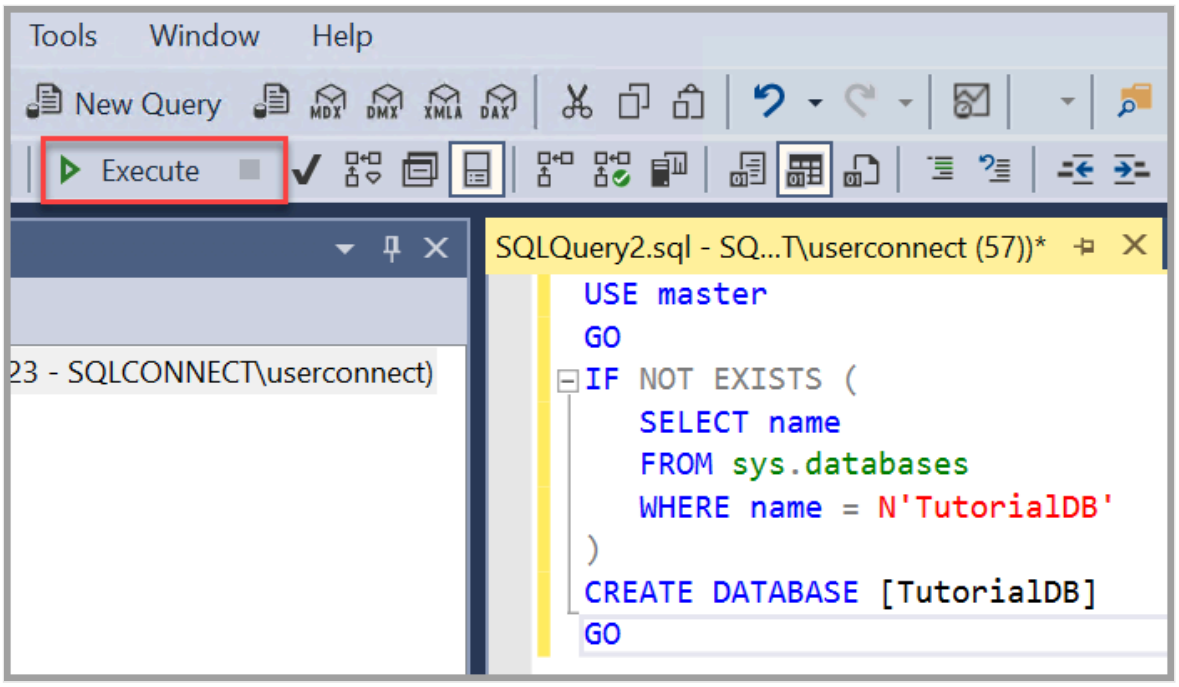


2. Paste the following T-SQL code snippet into the query window:

```SQL
IF NOT EXISTS (
    SELECT name
    FROM sys.databases
    WHERE name = N'TutorialDB'
)
CREATE DATABASE [TutorialDB];
GO
```

```
ALTER DATABASE [TutorialDB] SET QUERY_STORE = ON;
GO
```

3. Execute the query by selecting **Execute** or selecting F5 on your keyboard.

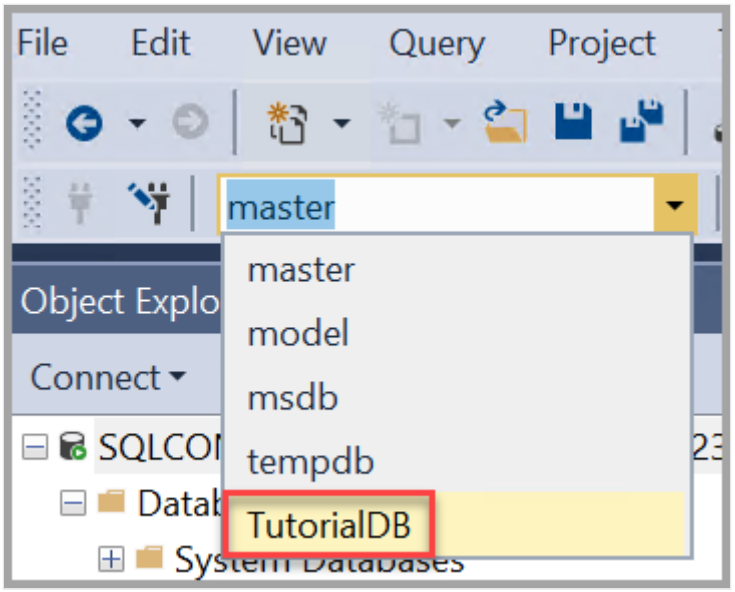


After the query is complete, the new `TutorialDB` database appears in the list of databases in Object Explorer. If it isn't displayed, right-click the **Databases** node, and then select **Refresh**.

Create a table in the new database

In this section, you create a table in the newly created `TutorialDB` database. Because the query editor is still in the context of the `master` database, switch the connection context to the `TutorialDB` database by doing the following steps:

1. In the database dropdown list, select the database that you want, as shown here:

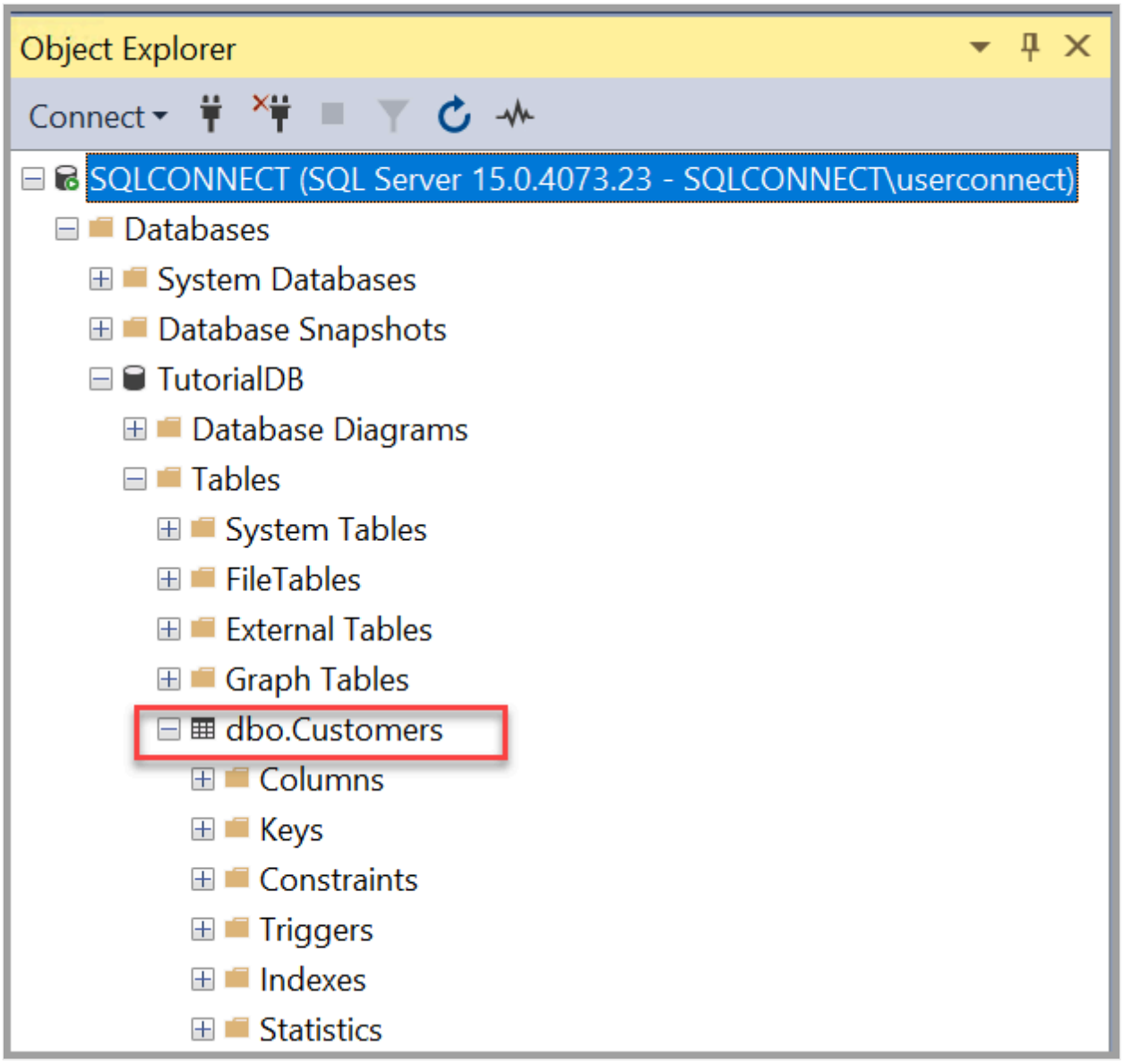


2. Paste the following T-SQL code snippet into the query window:

```SQL
USE [TutorialDB]
GO

-- Create a new table called 'Customers' in schema 'dbo'
-- Drop the table if it already exists
IF OBJECT_ID('dbo.Customers', 'U') IS NOT NULL
    DROP TABLE dbo.Customers
GO

-- Create the table in the specified schema
CREATE TABLE dbo.Customers (
    CustomerId INT NOT NULL PRIMARY KEY, -- primary key column
    Name NVARCHAR(50) NOT NULL,
    Location NVARCHAR(50) NOT NULL,
    Email NVARCHAR(50) NOT NULL
);
GO
```

3. Execute the query by selecting **Execute** or selecting F5 on your keyboard.

After the query is complete, the new Customers table is displayed in the list of tables in Object Explorer. If the table isn't displayed, right-click the **TutorialDB** > **Tables** node in Object Explorer, then select **Refresh**.



Insert rows into the new table

Now let's insert some rows into the Customers table that you created. Paste the following T-SQL code snippet into the query window, and then select **Execute**:

SQL

```
-- Insert rows into table 'Customers'
INSERT INTO dbo.Customers (
    CustomerId,
    Name,
    Location,
    Email
)
VALUES
    (1, N'Orlando', N'Australia', N''),
    (2, N'Keith', N'India', N'keith0@adventure-works.com'),
    (3, N'Donna', N'Germany', N'donna0@adventure-works.com'),
```

```
    (4, N'Janet', N'United States', N'janet1@adventure-works.com')
GO
```

Query the table and view the results

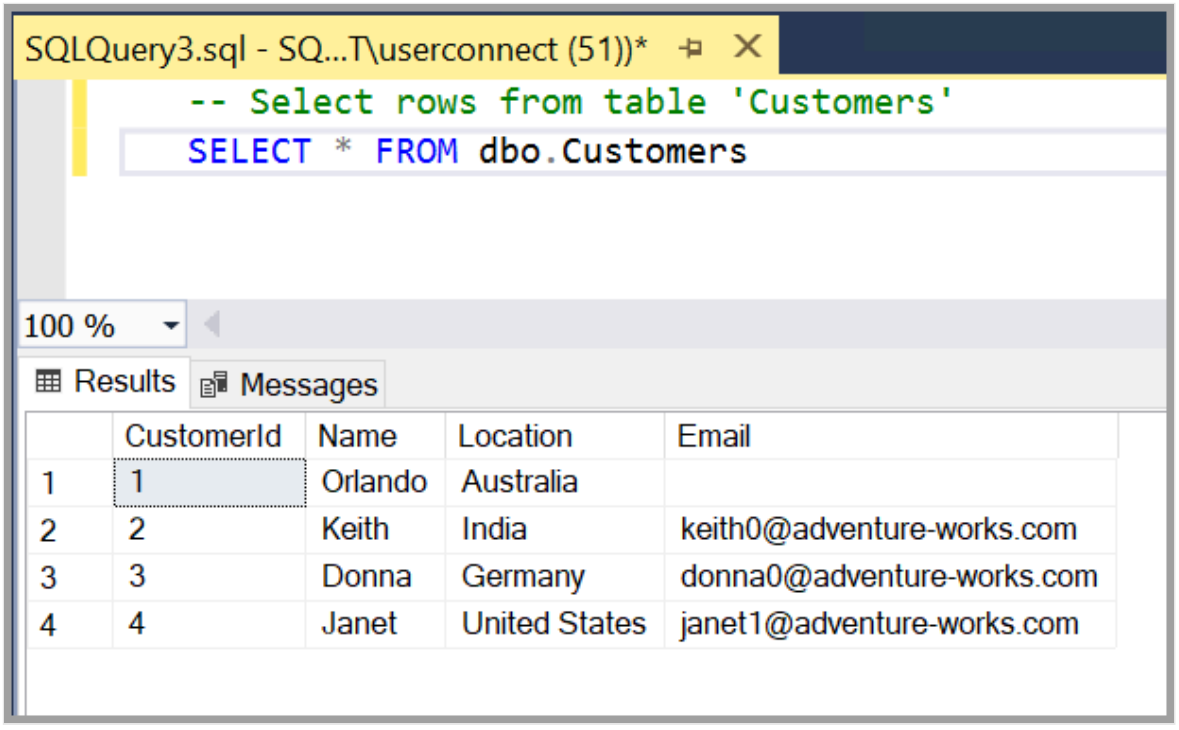
The results of a query are visible beneath the query text window. To query the `Customers` table and view the rows that were inserted, follow these steps:

1. Paste the following T-SQL code snippet into the query window, and then select **Execute**:

   SQL

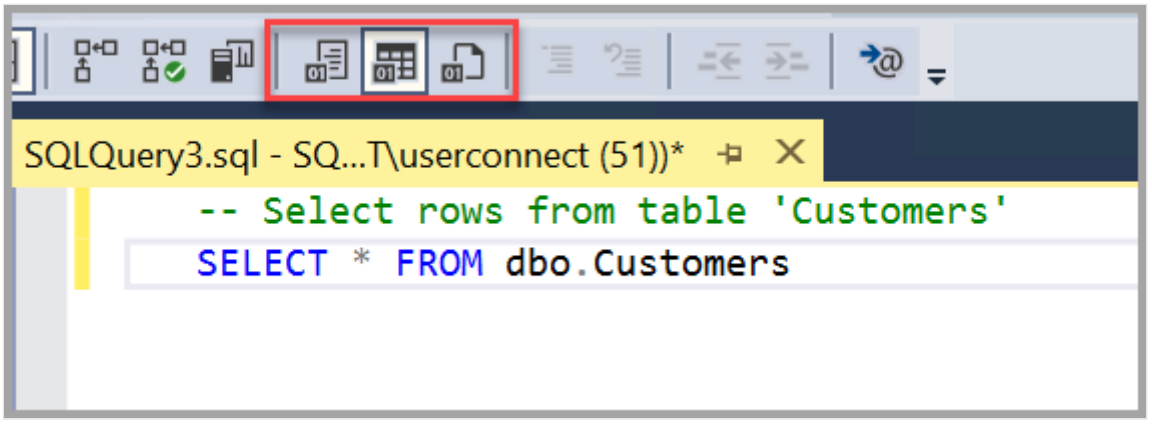
   ```
   -- Select rows from table 'Customers'
   SELECT * FROM dbo.Customers;
   ```

   The query results are displayed under the area where the text was entered.

	CustomerId	Name	Location	Email
1	1	Orlando	Australia	
2	2	Keith	India	keith0@adventure-works.com
3	3	Donna	Germany	donna0@adventure-works.com
4	4	Janet	United States	janet1@adventure-works.com

   You can also modify the way results are presented by selecting one of the following options:

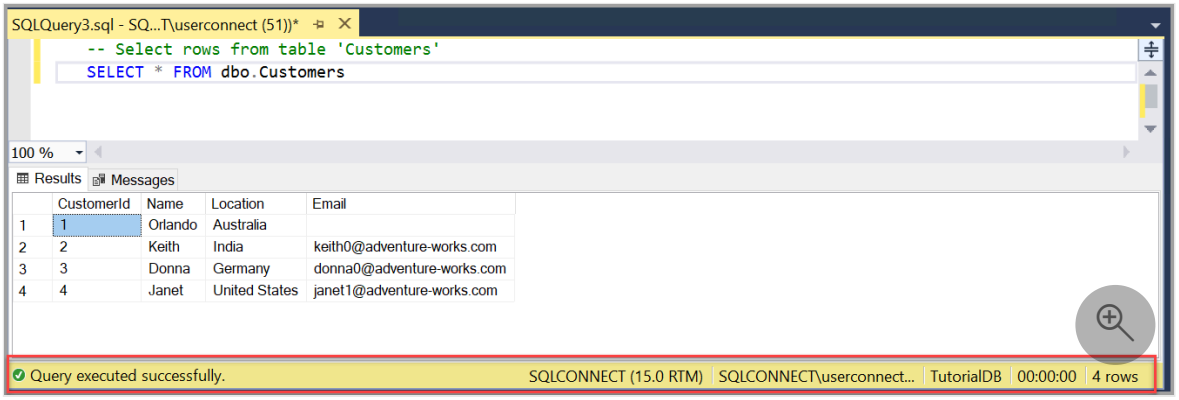


- The first button displays the results in **Text View**, as shown in the image in the next section.
- The middle button displays the results in **Grid View**, which is the default option.
- The third button lets you save the results to a file whose extension is .rpt by default.

Verify your connection properties by using the query window table

You can find information about the connection properties under the results of your query. After you run the previously mentioned query in the preceding step, review the connection properties at the bottom of the query window.

- You can determine which server and database you're connected to, and your username.
- You can also view the query duration and the number of rows the previously executed query returns.



Additional tools

You can also use Azure Data Studio to connect and query SQL Server, an Azure SQL Database, and Azure Synapse Analytics.

Related content

- SQL Server Management Studio (SSMS) Query Editor
- Script objects in SQL Server Management Studio
- Use templates in SQL Server Management Studio
- SQL Server Management Studio components and configuration
- Tips and tricks for using SQL Server Management Studio (SSMS)

Feedback

Was this page helpful? Yes No

Quickstart: Connect and query an Azure SQL Database or an Azure SQL Managed Instance using SQL Server Management Studio (SSMS)

Article • 02/13/2025

Applies to: Azure SQL Database

Get started using SQL Server Management Studio (SSMS) to connect to your Azure SQL Database and run some Transact-SQL (T-SQL) commands.

> **Note**
>
> While Microsoft Entra ID is the new name for Azure Active Directory (Azure AD), to prevent disrupting existing environments, Azure AD still remains in some hardcoded elements such as UI fields, connection providers, error codes, and cmdlets. In this article, the two names are interchangeable.

The article demonstrates the following steps:

- ✓ Connect to an Azure SQL database
- ✓ Create a database
- ✓ Create a table in your new database
- ✓ Insert rows into your new table
- ✓ Query the new table and view the results
- ✓ Use the query window table to verify your connection properties

Prerequisites

- SQL Server Management Studio (SSMS)
- Azure SQL Database or Azure SQL Managed Instance

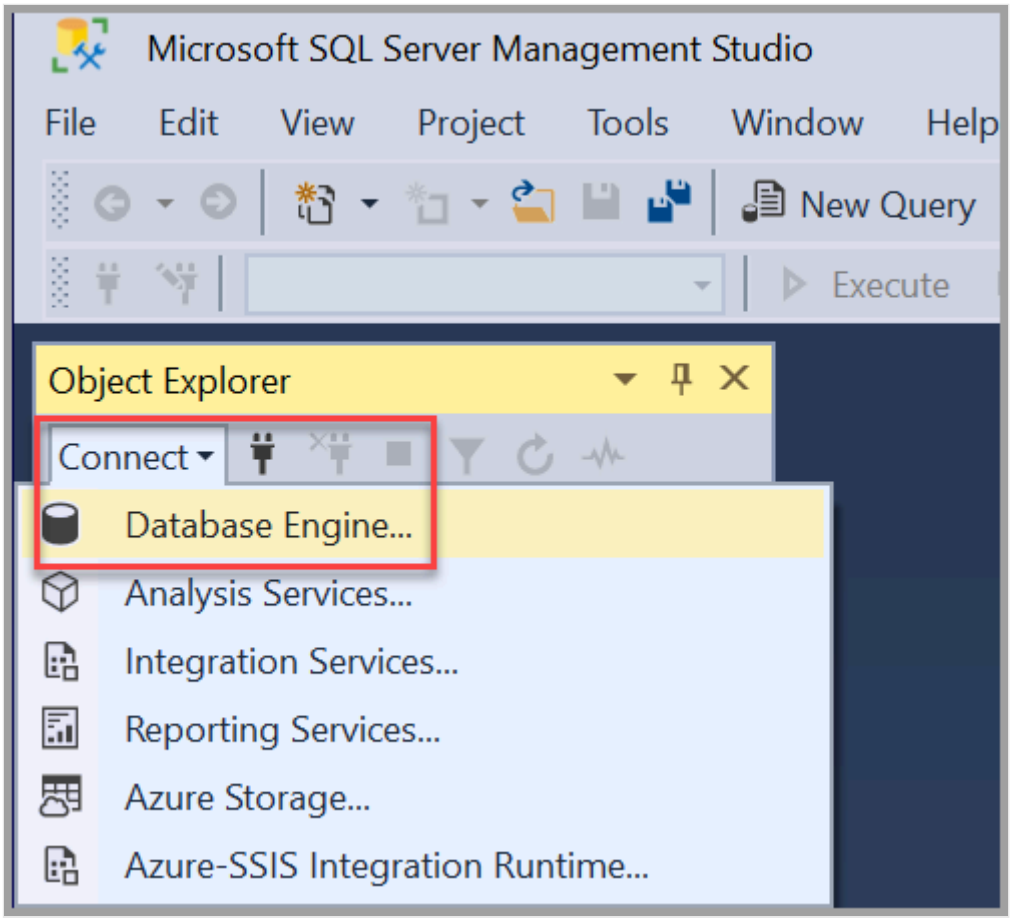
Connect to an Azure SQL Database or Azure SQL Managed Instance

> **Note**

In December 2021, releases of SSMS prior to 18.6 will no longer authenticate to Database Engines through Microsoft Entra multifactor authentication (MFA). To continue utilizing MFA, you need SSMS 18.6 or later.

Connectivity to Azure Analysis Services through Microsoft Entra multifactor authentication requires SSMS 18.5.1 or later.

1. Start SQL Server Management Studio (SSMS). The first time you run SSMS, the **Connect to Server** window opens. If it doesn't open, you can open it manually by selecting **Object Explorer** > **Connect** > **Database Engine**.



2. The **Connect to Server** dialog box appears. Enter the following information:

Expand table

Setting	Suggested value	Details
Server type	Database Engine	Select **Database Engine** (usually the default option).
Server name	The fully qualified server name	Enter the name of your *Azure SQL Database* or *Azure SQL Managed Instance* name.
Authentication		

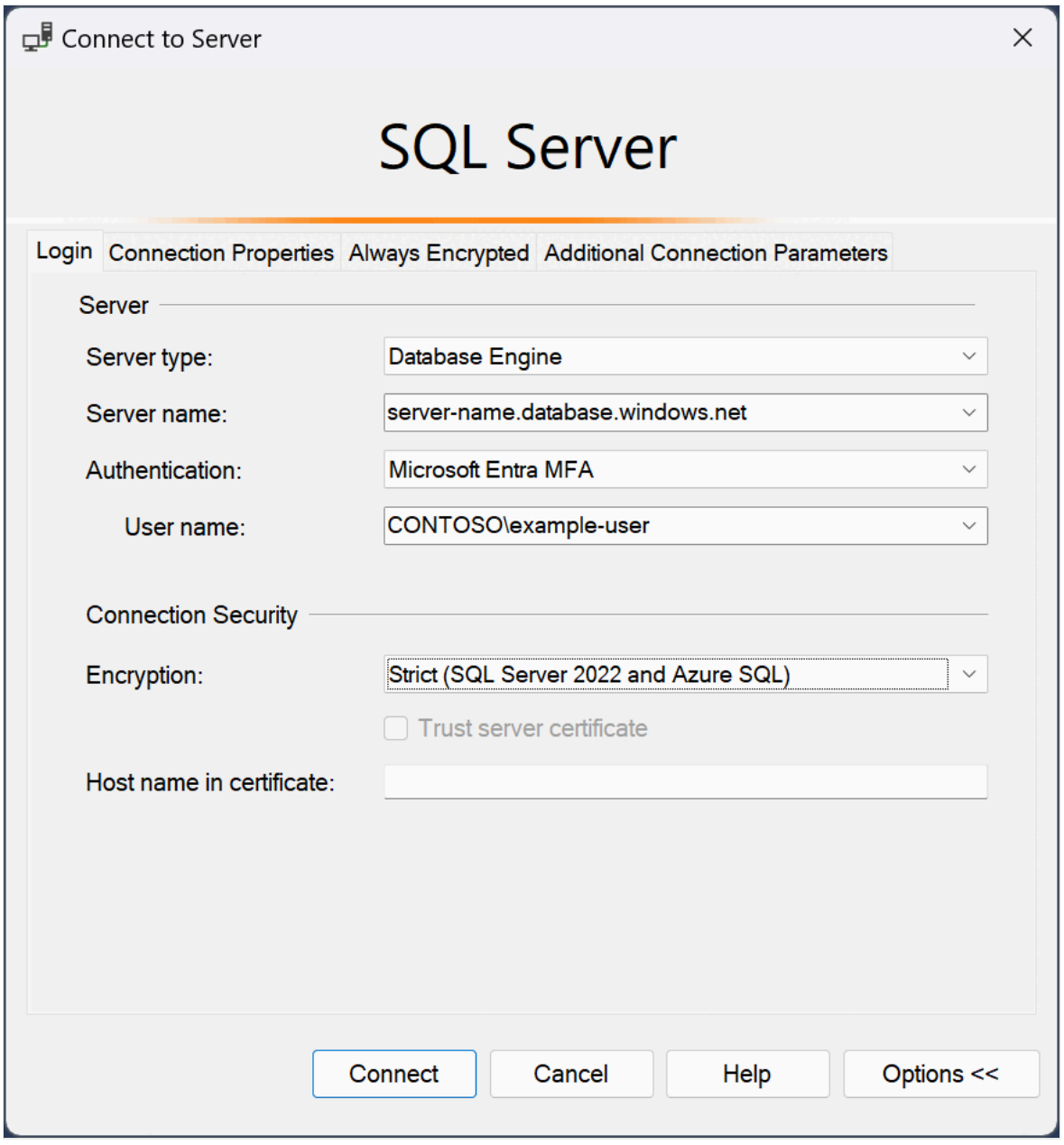
Setting	Suggested value	Details
	Microsoft Entra ID [1]	
	- Universal with MFA	See Using Microsoft Entra multifactor authentication.
	- Password - Integrated - Service Principal	See Microsoft Entra service principal with Azure SQL.
	- Managed Identity	See Managed identities in Microsoft Entra for Azure SQL. Connecting to a SQL instance with SSMS using a managed identity requires an Azure VM. See Use a Windows VM system-assigned managed identity to access Azure SQL
	- Default	The default option can be used when connecting using any Microsoft Entra authentication mode that's passwordless and noninteractive.
	SQL Server Authentication	Use **SQL Server Authentication** for Azure SQL to connect.
Login	Server account user ID	The user ID from the server account used to create the server.
Password	Server account password	The password from the server account used to create the server.
Encryption [2]	Encryption method	Select the encryption level for the connection. The default value is *Mandatory*.
Trust server certificate	Trust Server Certificate	Check this option to bypass server certificate validation. The default value is *False* (unchecked), which promotes better security using trusted certificates.
Host Name in Certificate	Host name of the server	The value provided in this option is used to specify a different, but expected, CN or SAN in the server certificate.

[1] The Windows Authentication method isn't supported for Azure SQL. For more information, see Azure SQL authentication.

[2] The default value is *Mandatory* in SSMS 20. *Strict (SQL Server 2022 and Azure SQL)* encryption should be used for Azure SQL Database and Azure SQL Managed

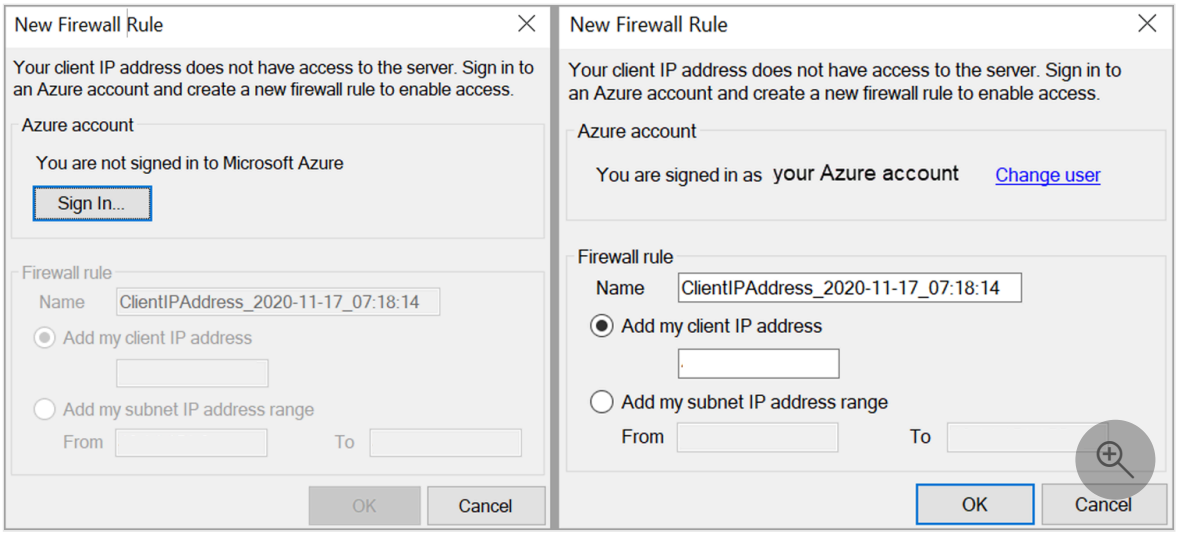
Instance. *Strict (SQL Server 2022 and Azure SQL)* encryption can be used for SQL Server when the instance has **Force Strict Encryption** enabled.

You can also modify additional connection options by selecting **Options**. Examples of connection options are the database you're connecting to, the connection timeout value, and the network protocol. This article uses the default values for all the options.

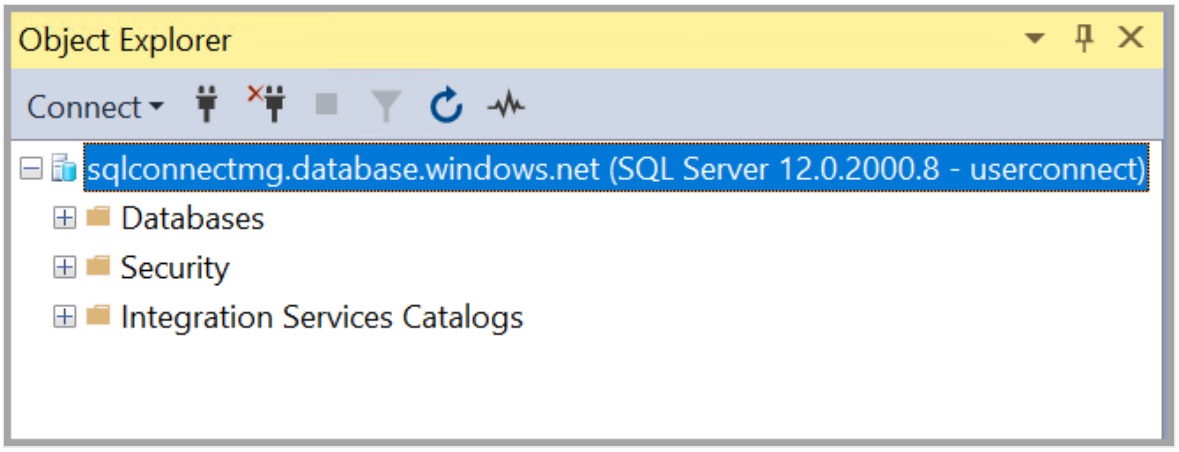


3. After you complete all the fields, select **Connect**.

   If your firewall isn't set up, a prompt appears to configure the firewall. Once you sign in, fill in your Azure account sign in information and continue to set the firewall rule. Then select **OK**. This prompt is a one time action. Once you configure the firewall, the firewall prompt shouldn't appear.



4. To verify that your Azure SQL Database or Azure SQL Managed Instance connection succeeded, expand and explore the objects within **Object Explorer** where the server name, the SQL Server version, and the username are displayed. These objects are different depending on the server type.



Troubleshoot connectivity issues

If your browser fails to connect to SQL Database when using **Microsoft Entra with MFA**, you can navigate to **Tools** > **Options** > **Azure Cloud**, and change the value for either **Use system default web browser** or **Use Web Account Manager**. For more information, see Options (Azure Services).

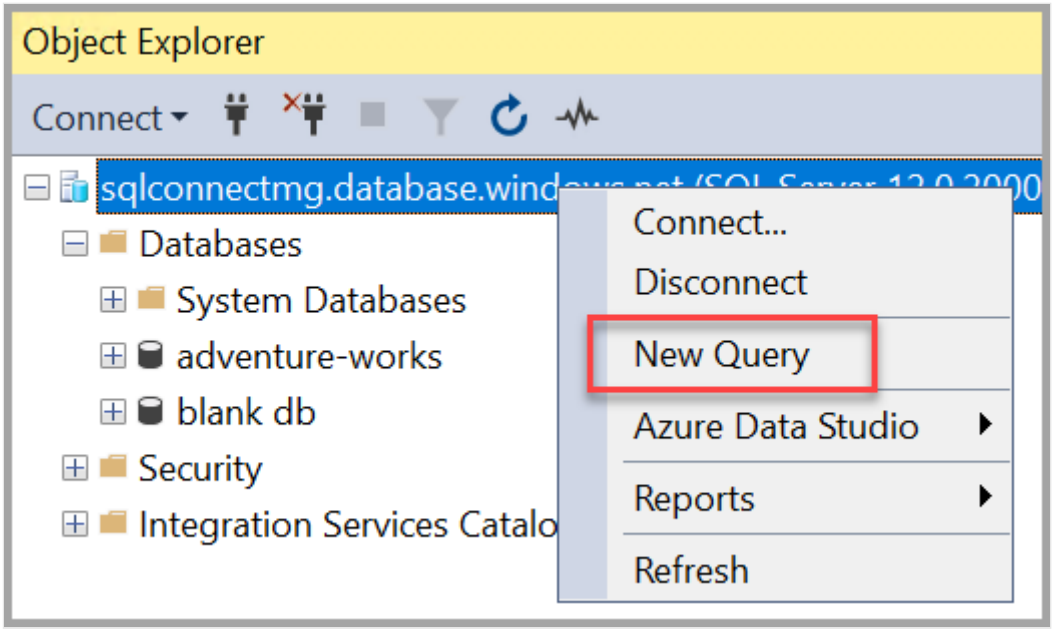
You can experience connection problems with Azure SQL Database. For more information on troubleshooting connection problems, visit Troubleshooting connectivity issues.

You can prevent, troubleshoot, diagnose, and mitigate connection and transient errors that you encounter when interacting with Azure SQL Database or Azure SQL Managed Instance. For more information, visit Troubleshoot transient connection errors.

Create a database

Now let's create a database named TutorialDB by following the below steps:

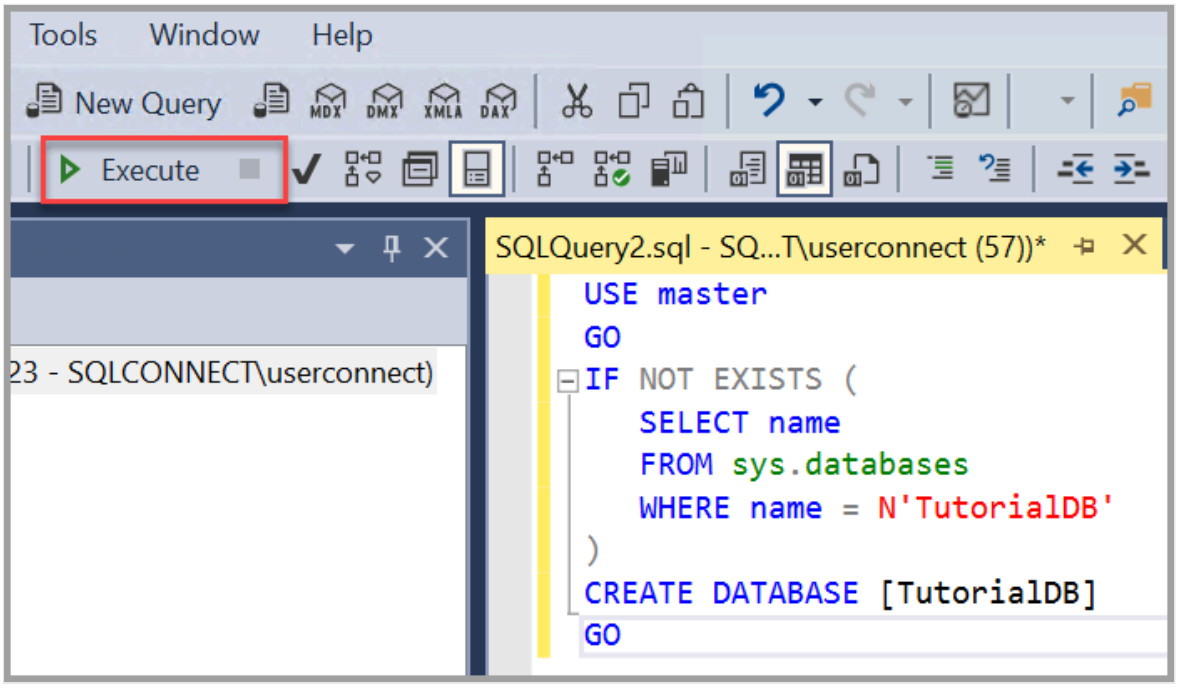
1. Right-click your server instance in Object Explorer, and then select **New Query**:



2. Paste the following T-SQL code snippet into the query window:

```SQL
IF NOT EXISTS (
   SELECT name
   FROM sys.databases
   WHERE name = N'TutorialDB'
)
CREATE DATABASE [TutorialDB];
GO

ALTER DATABASE [TutorialDB]
SET QUERY_STORE = ON;
GO
```

3. Execute the query by selecting **Execute** or selecting F5 on your keyboard.

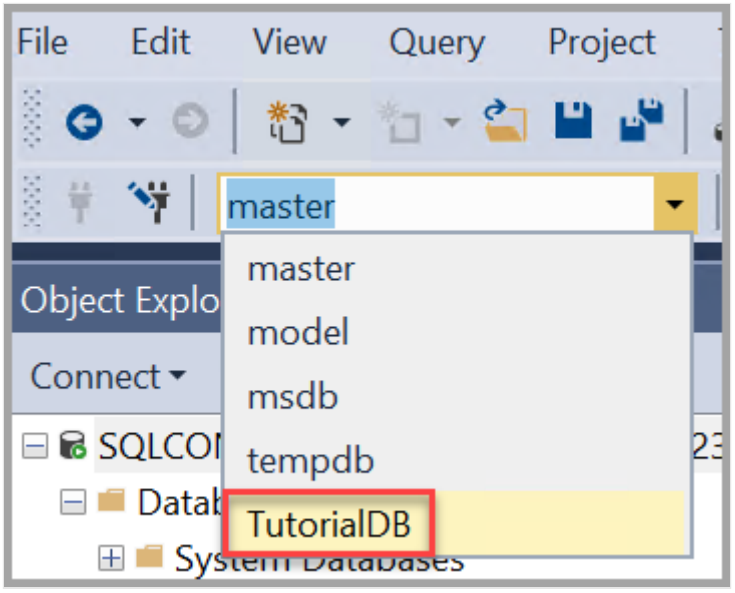


After the query is complete, the new TutorialDB database appears in the list of databases in Object Explorer. If it isn't displayed, right-click the **Databases** node, and then select **Refresh**.

Create a table in the new database

In this section, you create a table in the newly created TutorialDB database. Because the query editor is still in the context of the `master` database, switch the connection context to the *TutorialDB* database by doing the following steps:

1. In the database dropdown list, select the database that you want, as shown here:



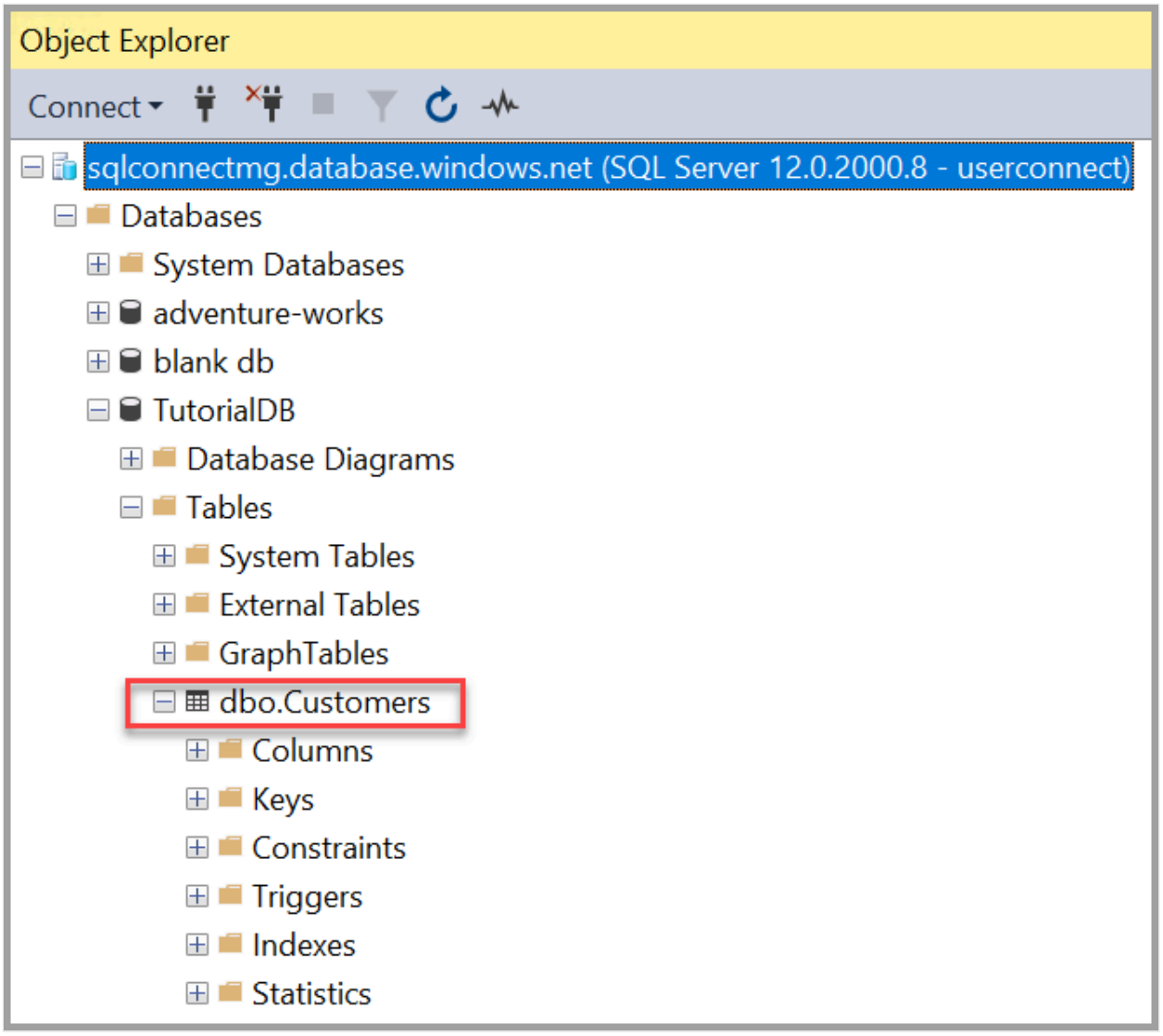
2. Paste the following T-SQL code snippet into the query window:

```
SQL
```

```
USE [TutorialDB];
GO

-- Create a new table called 'Customers' in schema 'dbo'
-- Drop the table if it already exists
IF OBJECT_ID('dbo.Customers', 'U') IS NOT NULL
    DROP TABLE dbo.Customers;
GO

-- Create the table in the specified schema
CREATE TABLE dbo.Customers (
    CustomerId INT NOT NULL PRIMARY KEY, -- primary key column
    Name NVARCHAR(50) NOT NULL,
    Location NVARCHAR(50) NOT NULL,
    Email NVARCHAR(50) NOT NULL
);
GO
```

3. Execute the query by selecting **Execute** or selecting F5 on your keyboard.

After the query is complete, the new Customers table is displayed in the list of tables in Object Explorer. If the table isn't displayed, right-click the **TutorialDB** > **Tables** node in Object Explorer, and then select **Refresh**.



Insert rows into the new table

Now let's insert some rows into the Customers table that you created. Paste the following T-SQL code snippet into the query window, and then select **Execute**:

SQL

```
-- Insert rows into table 'Customers'
INSERT INTO dbo.Customers (
    [CustomerId],
    [Name],
    [Location],
    [Email]
)
VALUES
   (1, N'Orlando', N'Australia', N''),
   (2, N'Keith', N'India', N'keith0@adventure-works.com'),
   (3, N'Donna', N'Germany', N'donna0@adventure-works.com'),
   (4, N'Janet', N'United States', N'janet1@adventure-works.com');
GO
```

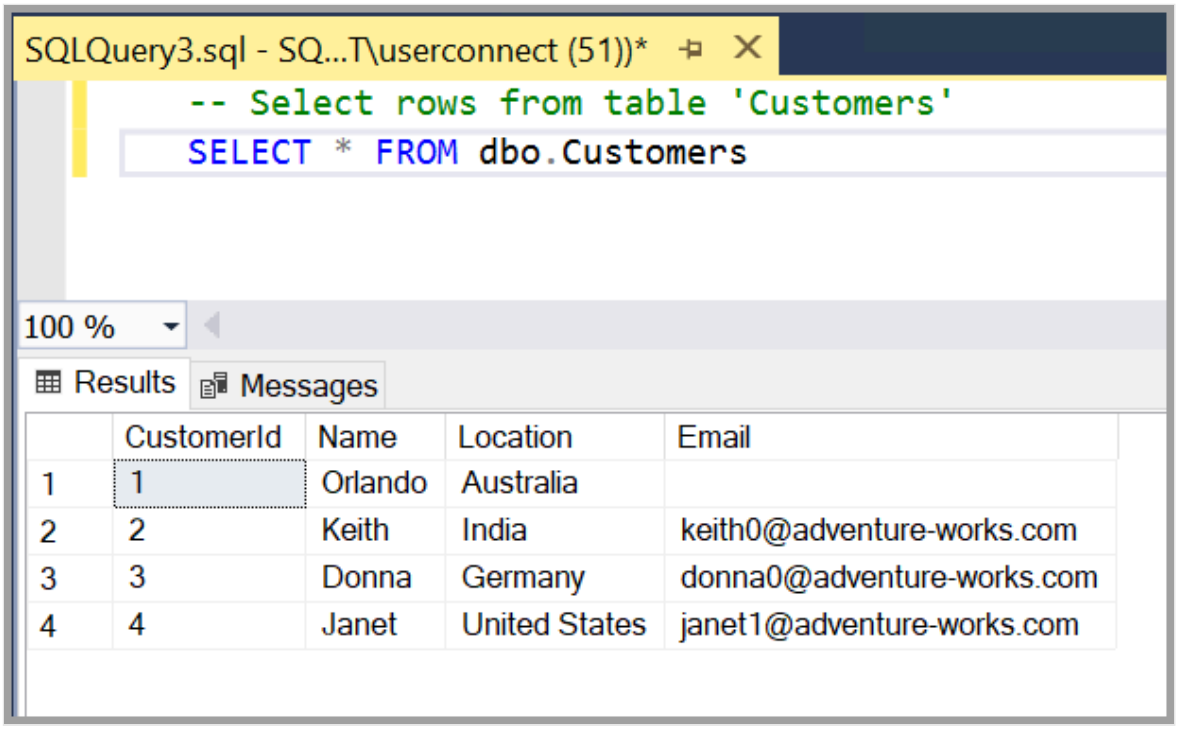
Query the table and view the results

The results of a query are visible beneath the query text window. To query the `Customers` table and view the rows that were inserted, follow these steps:

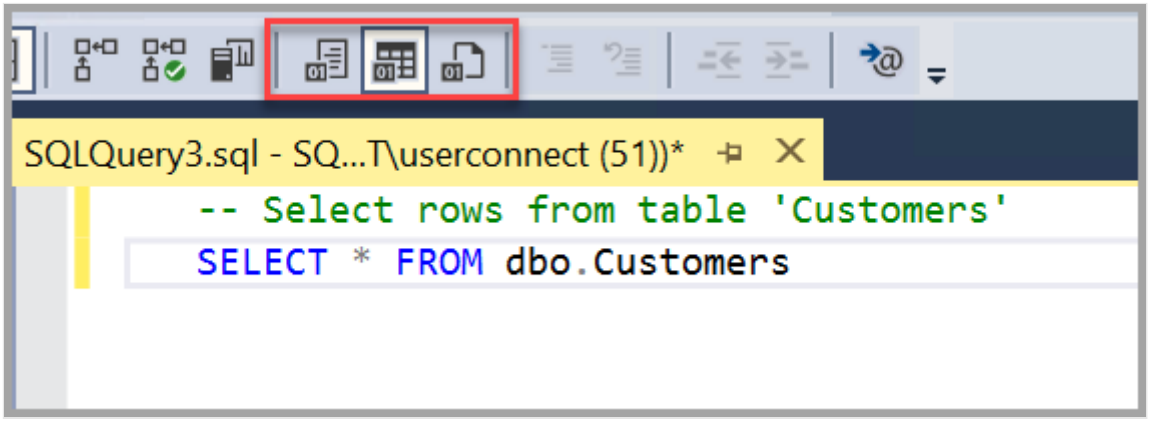
1. Paste the following T-SQL code snippet into the query window, and then select **Execute**:

```SQL
-- Select rows from table 'Customers'
SELECT * FROM dbo.Customers;
```

The query results are displayed under the area where the text was entered.



You can also modify the way results are presented by selecting one of the following options:



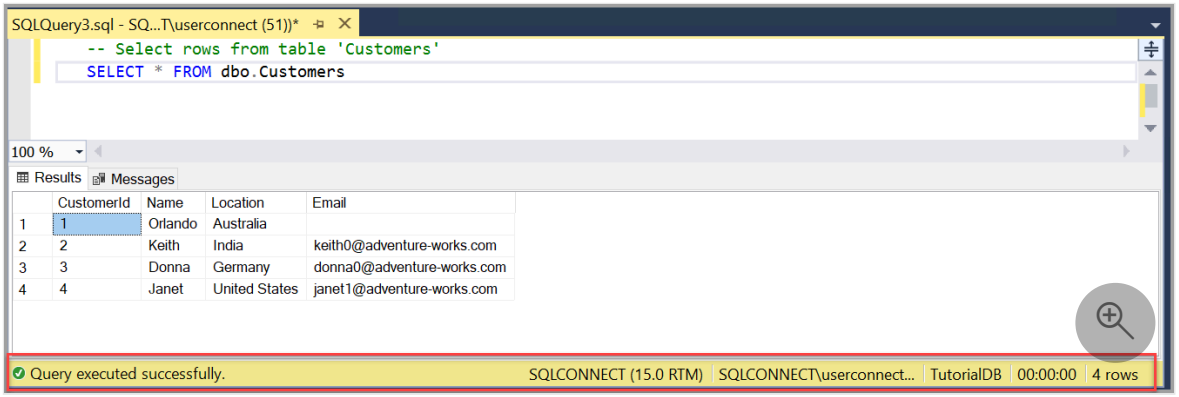
- The first button displays the results in **Text View**, as shown in the image in the next section.
- The middle button displays the results in **Grid View**, which is the default option.
- The third button lets you save the results to a file whose extension is .rpt by default.

Verify your connection properties by using the query window table

You can find information about the connection properties under the results of your query. After you run the previously mentioned query in the preceding step, review the connection properties at the bottom of the query window.

- You can determine which server and database you're connected to, and your username.

- You can also view the query duration and the number of rows returned by the previously executed query.



Additional tools

You can also use Azure Data Studio to connect and query SQL Server, Azure SQL Database, and Azure Synapse Analytics.

Related content

- SQL Server Management Studio (SSMS) Query Editor
- Script objects in SQL Server Management Studio
- Use templates in SQL Server Management Studio
- SQL Server Management Studio components and configuration

- Tips and tricks for using SQL Server Management Studio (SSMS)

Feedback

Was this page helpful? Yes No

Quickstart: Connect and query a dedicated SQL pool (formerly SQL DW) in Azure Synapse Analytics with SQL Server Management Studio (SSMS)

Article • 02/13/2025

Applies to: Azure Synapse Analytics

In this quickstart, you can get started using SQL Server Management Studio (SSMS) to connect to your dedicated SQL pool (formerly SQL DW) in Azure Synapse Analytics and run some Transact-SQL (T-SQL) commands.

- ✓ Connect to a dedicated SQL pool (formerly SQL DW) in Azure Synapse Analytics
- ✓ Create a table in your new database
- ✓ Insert rows into your new table
- ✓ Query the new table and view the results
- ✓ Use the query window table to verify your connection properties

Prerequisites

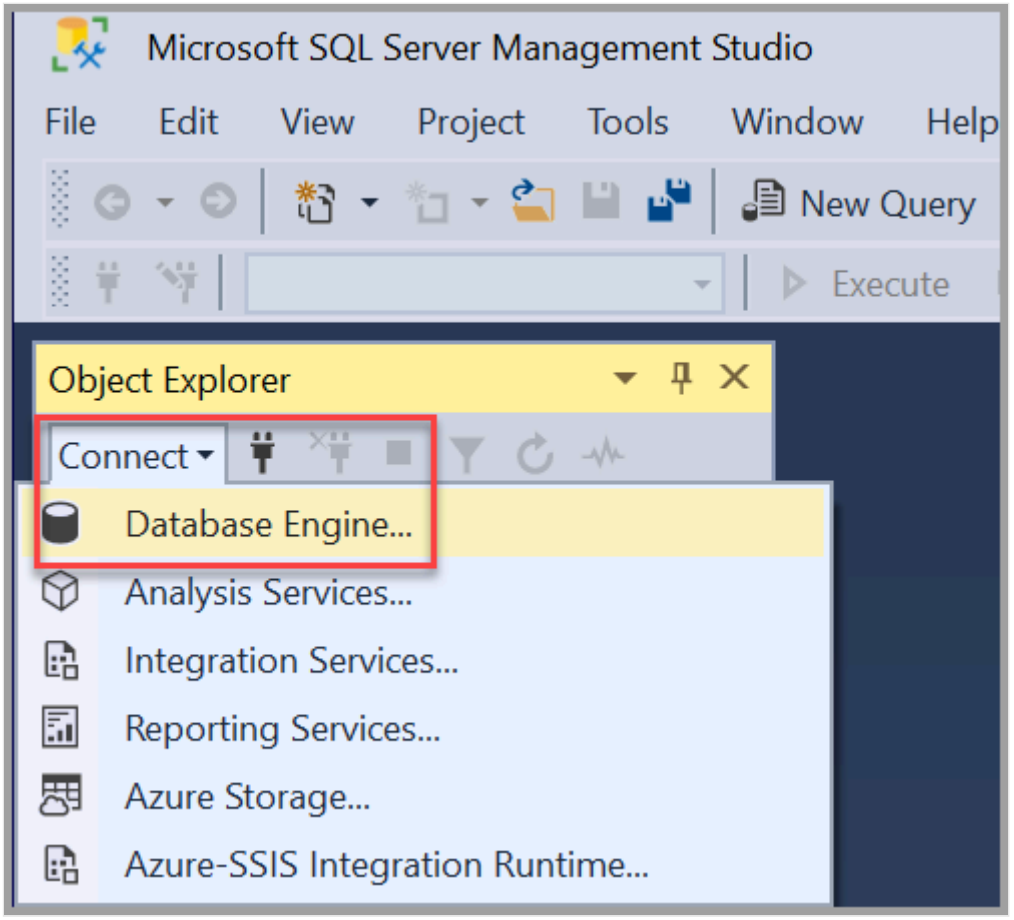
To complete this article, you need SQL Server Management Studio (SSMS) and access to a data source.

- Install SQL Server Management Studio
- Azure Synapse Analytics

Connect to a dedicated SQL pool (formerly SQL DW) in Azure Synapse Analytics

> **Note**
>
> In December 2021, releases of SSMS prior to 18.6 will no longer authenticate to Database Engines through Microsoft Entra multifactor authentication (MFA). To continue utilizing MFA, you need SSMS 18.6 or later.
>
> Connectivity to Azure Analysis Services through Microsoft Entra multifactor authentication requires SSMS 18.5.1 or later.

1. Start SQL Server Management Studio. The first time you run SSMS, the **Connect to Server** window opens. If it doesn't open, you can open it manually by selecting **Object Explorer** > **Connect** > **Database Engine**.



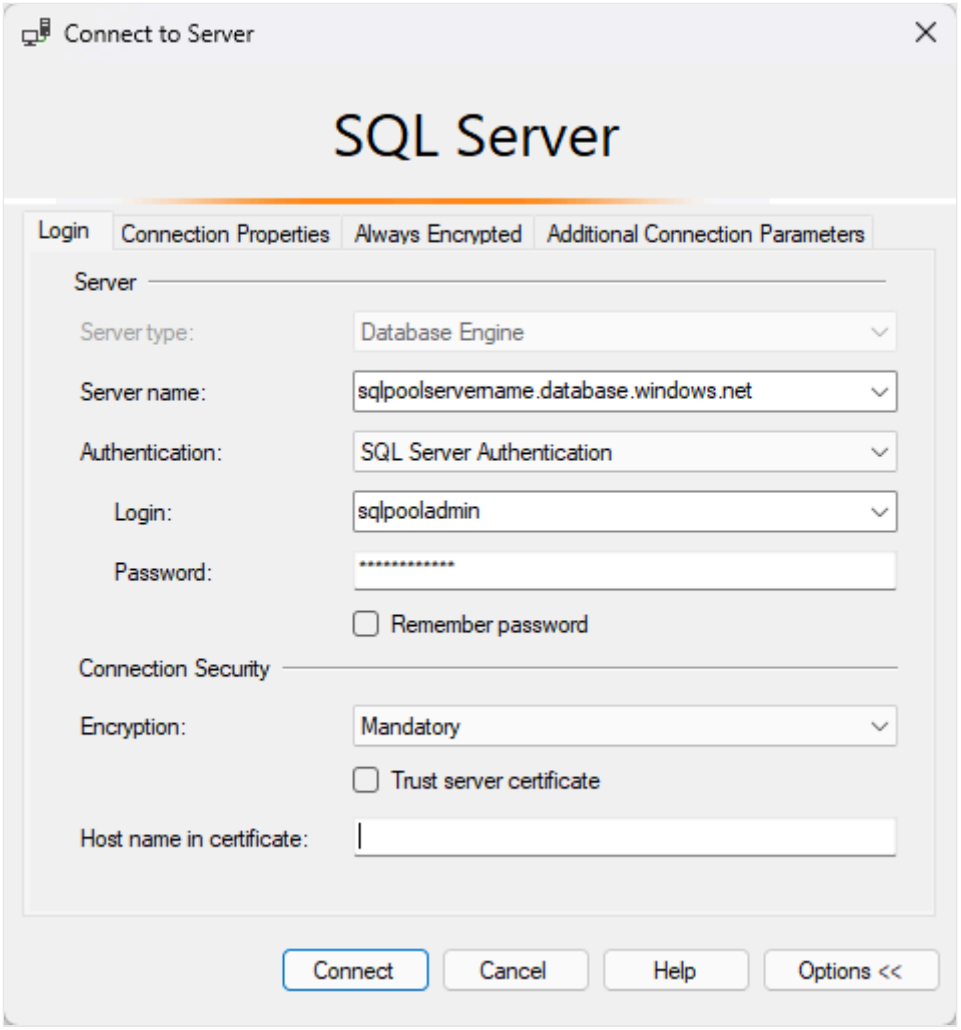
2. In the **Connect to Server** window, use the following list for guidance:

Expand table

Setting	Suggested values	Description
Server type	Database engine	For **Server type**, select **Database Engine** (usually the default option).
Server name	The fully qualified server name	For **Server name**, enter the name of your dedicated SQL pool (formerly SQL DW) server name.
Authentication	SQL Server Authentication	Use **SQL Server Authentication** to connect to a dedicated SQL pool (formerly SQL DW). The **Windows Authentication** method isn't supported for Azure SQL. For more information, see Azure SQL authentication.
Login	Server account user ID	The user ID from the server account used to create the server.

Setting	Suggested values	Description
Password	Server account password	The password from the server account used to create the server.
Encryption [1]	Encryption method	Select the encryption level for the connection. The default value is *Mandatory*.
Trust server certificate	Trust Server Certificate	Check this option to bypass server certificate validation. The default value is *False* (unchecked), which promotes better security using trusted certificates.
Host Name in Certificate	Host name of the server	The value provided in this option is used to specify a different, but expected, CN or SAN in the server certificate.

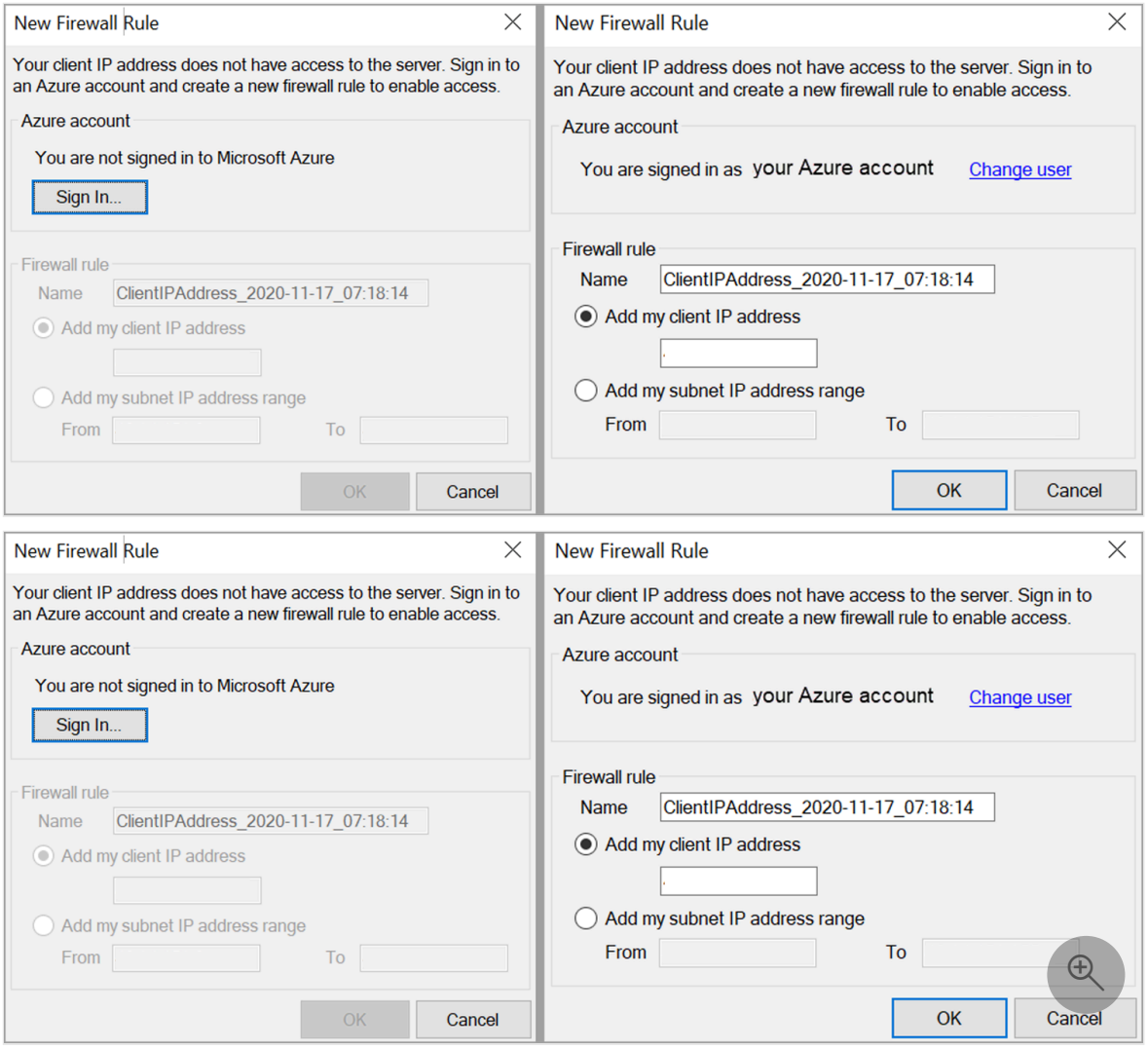
[1] The default value is *Mandatory* in SSMS 20. *Strict (SQL Server 2022 and Azure SQL)* encryption should be used for Azure SQL Database and Azure SQL Managed Instance. *Strict (SQL Server 2022 and Azure SQL)* encryption can be used for SQL Server when the instance has **Force Strict Encryption** enabled.



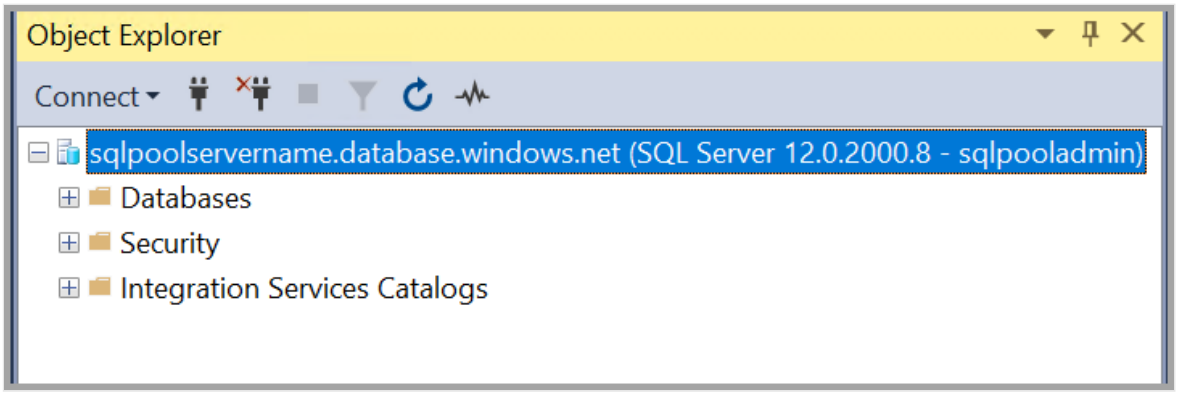
3. After you complete all the fields, select **Connect**.

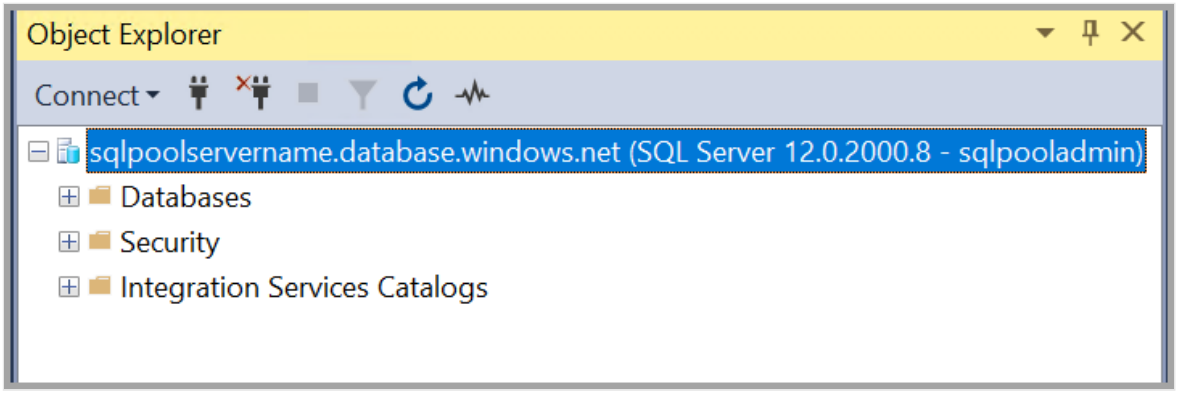
You can also modify other connection options by selecting **Options**. Examples of connection options are the database you're connecting to, the connection timeout value, and the network protocol. This article uses the default values for all the options.

If your firewall isn't set up, a prompt appears to configure the firewall. Once you sign in, fill in your Azure account sign in information and continue to set the firewall rule. Then select **OK**. This prompt is a one time action. Once you configure the firewall, the firewall prompt shouldn't appear.



4. To verify that your dedicated SQL pool (formerly SQL DW) connection succeeded, expand and explore the objects within **Object Explorer** where the server name, the SQL Server version, and the username are displayed. These objects are different depending on the server type.





Troubleshoot connectivity issues

You can experience connection problems with dedicated SQL pool (formerly SQL DW). For more information on troubleshooting connection problems, visit Troubleshooting connectivity issues.

Create a table

In this section, you create a table in your dedicated SQL pool (formerly SQL DW).

1. In Object Explorer, right-click on your dedicated SQL pool (formerly SQL DW), select **New query**.

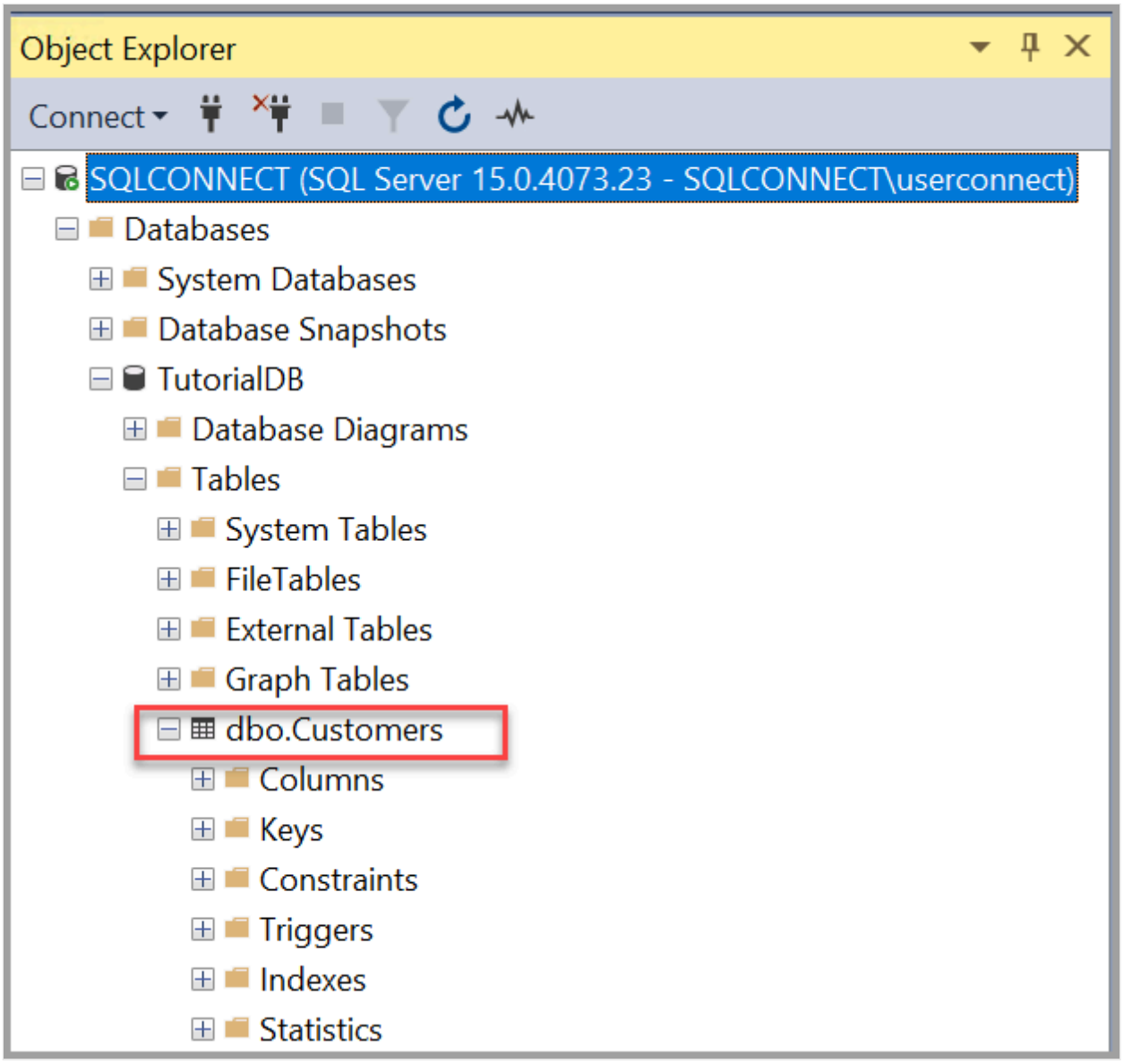
2. Paste the following T-SQL code snippet into the query window:

```SQL
-- Create a new table called 'Customers' in schema 'dbo'
-- Drop the table if it already exists
IF OBJECT_ID('dbo.Customers', 'U') IS NOT NULL
    DROP TABLE dbo.Customers
GO

-- Create the table in the specified schema
CREATE TABLE dbo.Customers (
    CustomerId INT NOT NULL,
    Name NVARCHAR(50) NOT NULL,
```

```
    Location NVARCHAR(50) NOT NULL,
    Email NVARCHAR(50) NOT NULL
);
GO
```

3. Execute the query by selecting **Execute** or selecting F5 on your keyboard.

After the query is complete, the new Customers table is displayed in the list of tables in Object Explorer. If the table isn't displayed, right-click the dedicated SQL pool (formerly SQL DW) **Tables** node in Object Explorer, and then select **Refresh**.



Insert rows into the new table

Now let's insert some rows into the Customers table that you created. Paste the following T-SQL code snippet into the query window, and then select **Execute**:

```
SQL
```

```
-- Insert rows into table 'Customers'
INSERT INTO dbo.Customers VALUES ( 1, N'Orlando', N'Australia', N'');
INSERT INTO dbo.Customers VALUES ( 2, N'Keith', N'India',
N'keith0@adventure-works.com');
INSERT INTO dbo.Customers VALUES (3, N'Donna', N'Germany',
N'donna0@adventure-works.com');
INSERT INTO dbo.Customers VALUES (4, N'Janet', N'United States',
N'janet1@adventure-works.com');
```

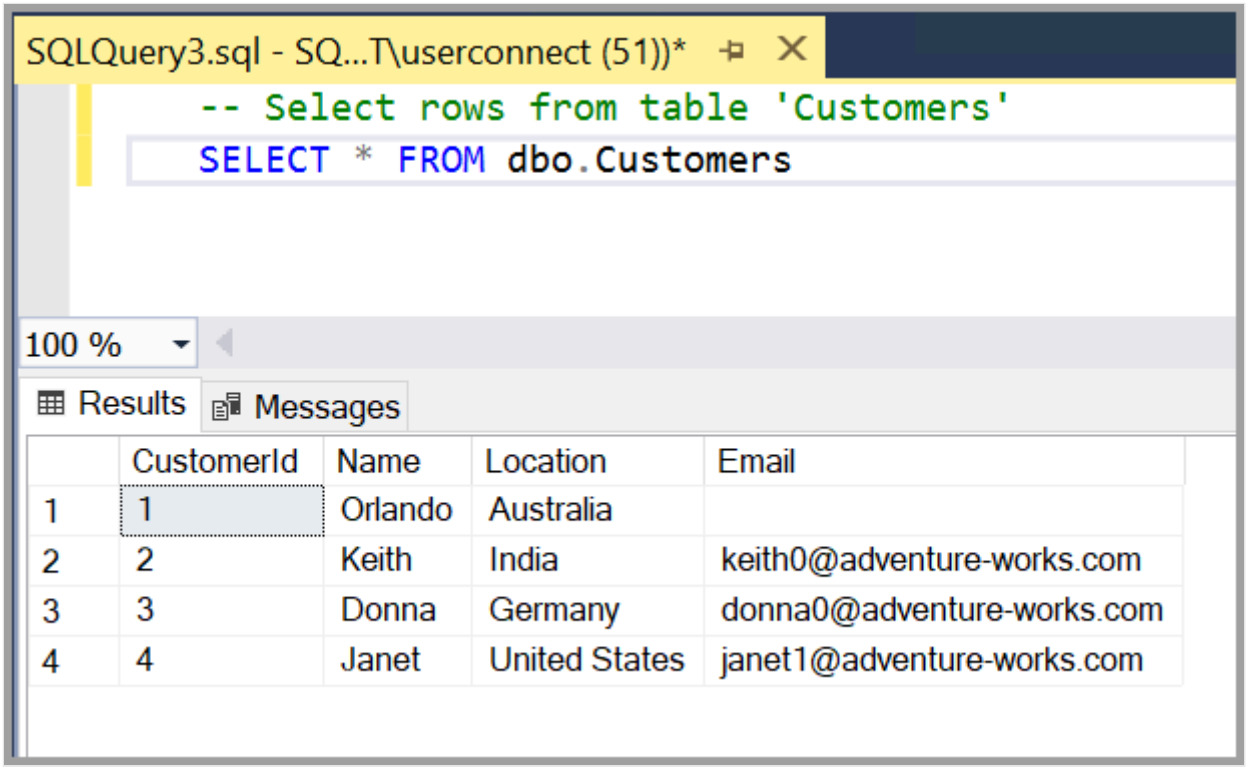
Query the table and view the results

The results of a query are visible beneath the query text window. To query the `Customers` table and view the rows that were inserted, paste the following T-SQL code snippet into the query window, and then select **Execute**:

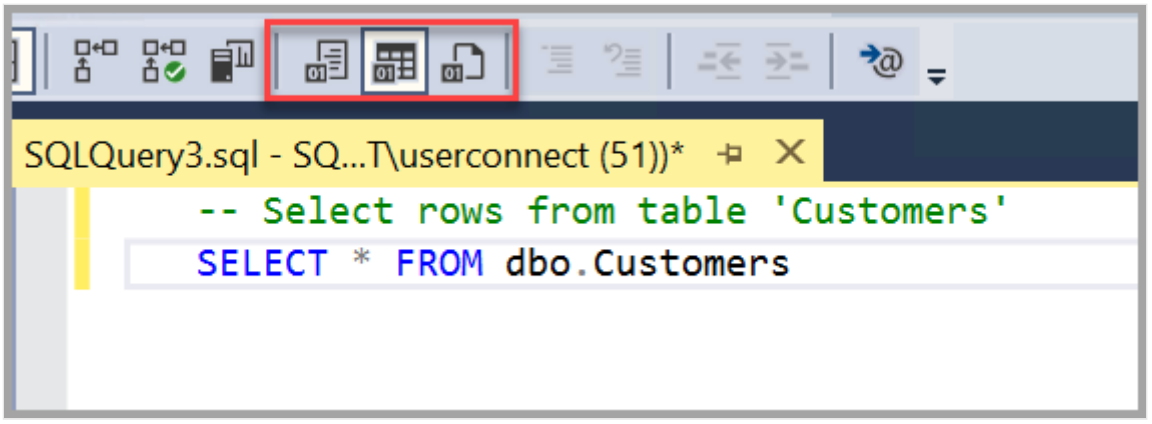
SQL

```
-- Select rows from table 'Customers'
SELECT * FROM dbo.Customers;
```

The query results are displayed under the area where the text was entered.



You can also modify the way results are presented by selecting one of the following options:



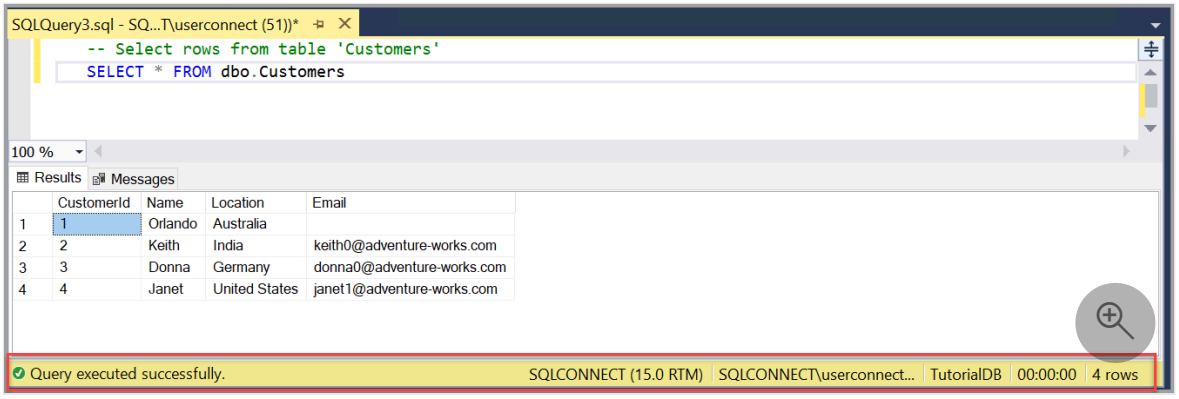
- The first button displays the results in **Text View**, as shown in the image in the next section.
- The middle button displays the results in **Grid View**, which is the default option.
- The third button lets you save the results to a file whose extension is `.rpt` by default.

Verify your connection properties by using the query window table

You can find information about the connection properties under the results of your query. After you run the previously mentioned query in the preceding step, review the connection properties at the bottom of the query window.

- You can determine which server and database you're connected to, and your username.

- You can also view the query duration and the number of rows returned by the previously executed query.



Additional tools

You can also use Azure Data Studio to connect and query SQL Server, an Azure SQL Database, and Azure Synapse Analytics.

Related content

- SQL Server Management Studio (SSMS) Query Editor
- Script objects in SQL Server Management Studio
- Use templates in SQL Server Management Studio
- SQL Server Management Studio components and configuration
- Tips and tricks for using SQL Server Management Studio (SSMS)

Feedback

Was this page helpful? Yes No

SQL Server Management Studio components and configuration

Article • 02/13/2025

This tutorial describes the various window components in SQL Server Management Studio (SSMS), and a set of basic configuration options for your workspace. In this article, you learn how to:

- ✓ Identify the components that make up the SSMS environment
- ✓ Change the environment layout, and reset it to the default
- ✓ Maximize the query editor
- ✓ Change the font
- ✓ Configure startup options

Prerequisites

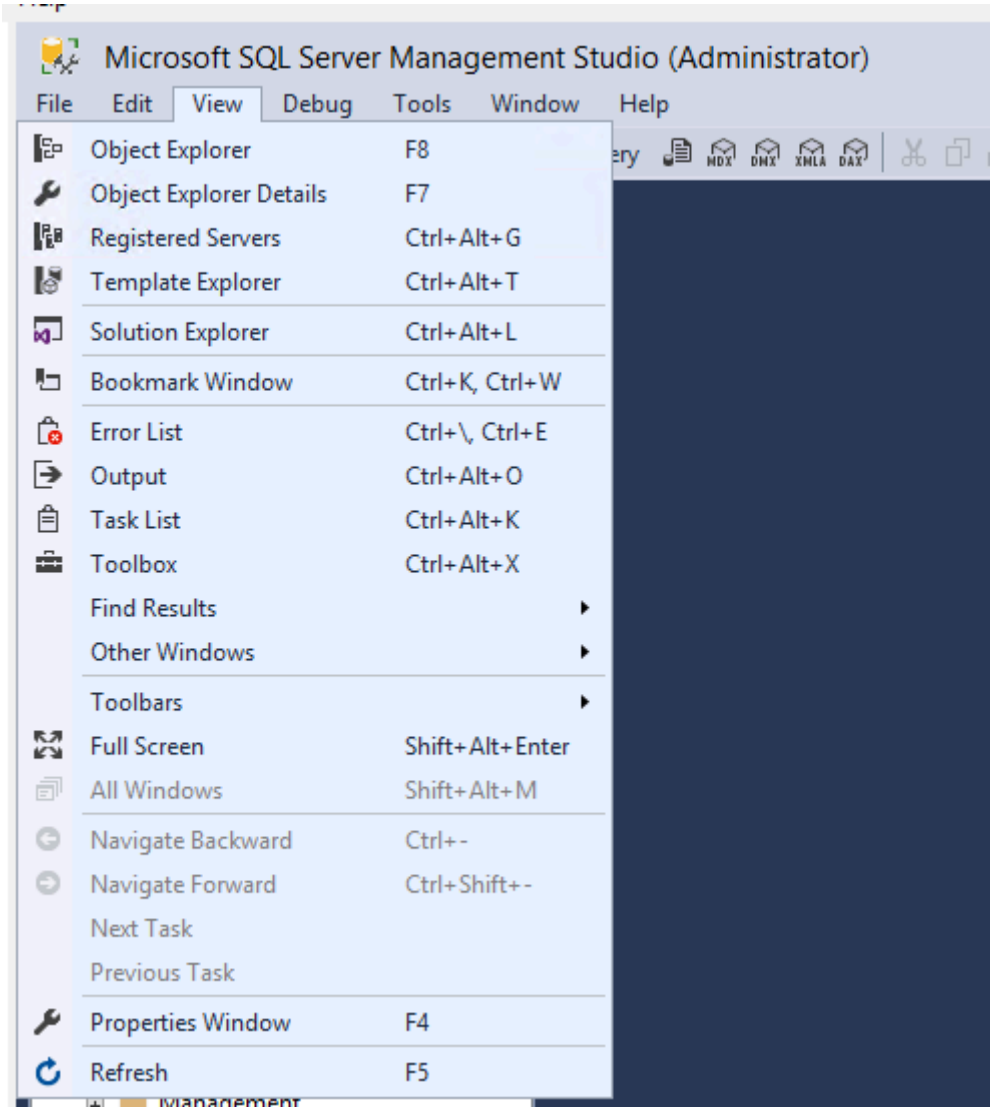
To complete this tutorial, you need SQL Server Management Studio.

- Install SQL Server Management Studio.

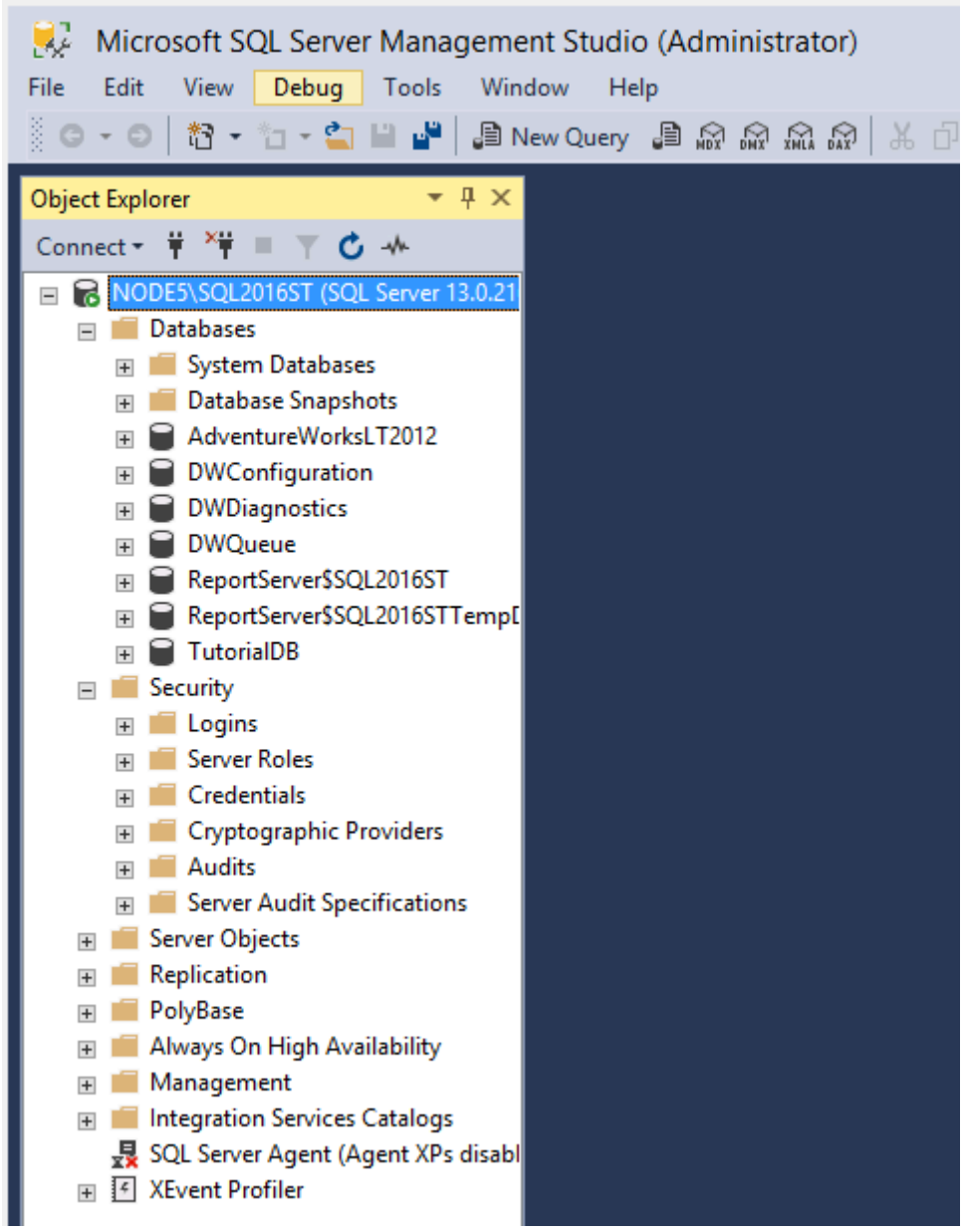
SQL Server Management Studio components

This section describes the various window components that are available in the workspace and how to use them.

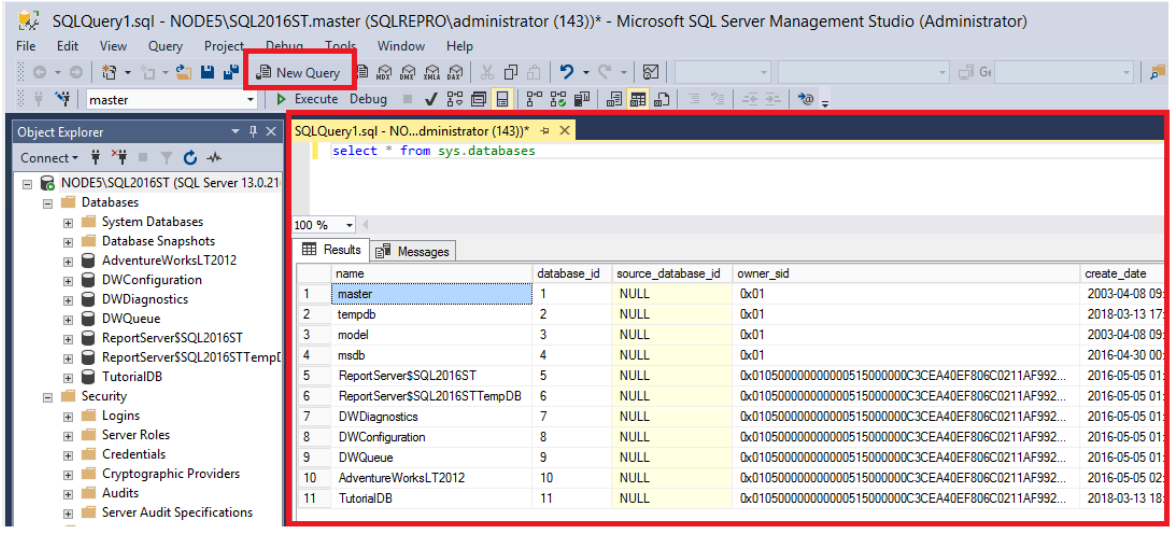
- To close a window, select the **X** in the right corner of the title bar.
- To reopen a window, select the window in the **View** menu.



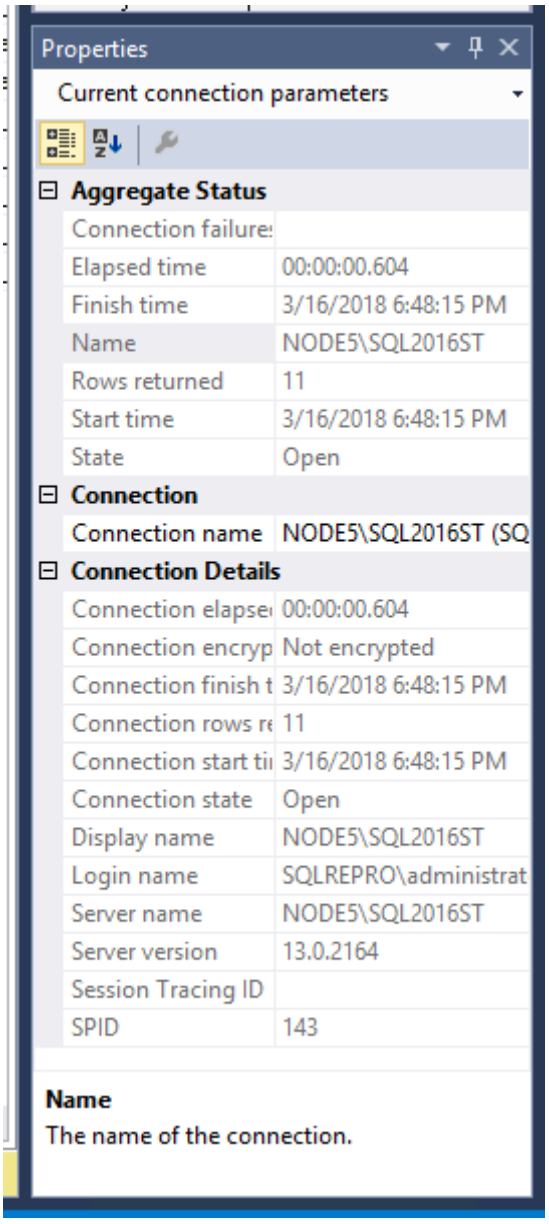
- **Object Explorer** (F8): Object Explorer is a tree view of all the objects for a database or server. This view includes the databases of the SQL Server Database Engine, SQL Server Analysis Services, SQL Server Reporting Services, and SQL Server Integration Services. Object Explorer displays information for all servers to which there's a connection.



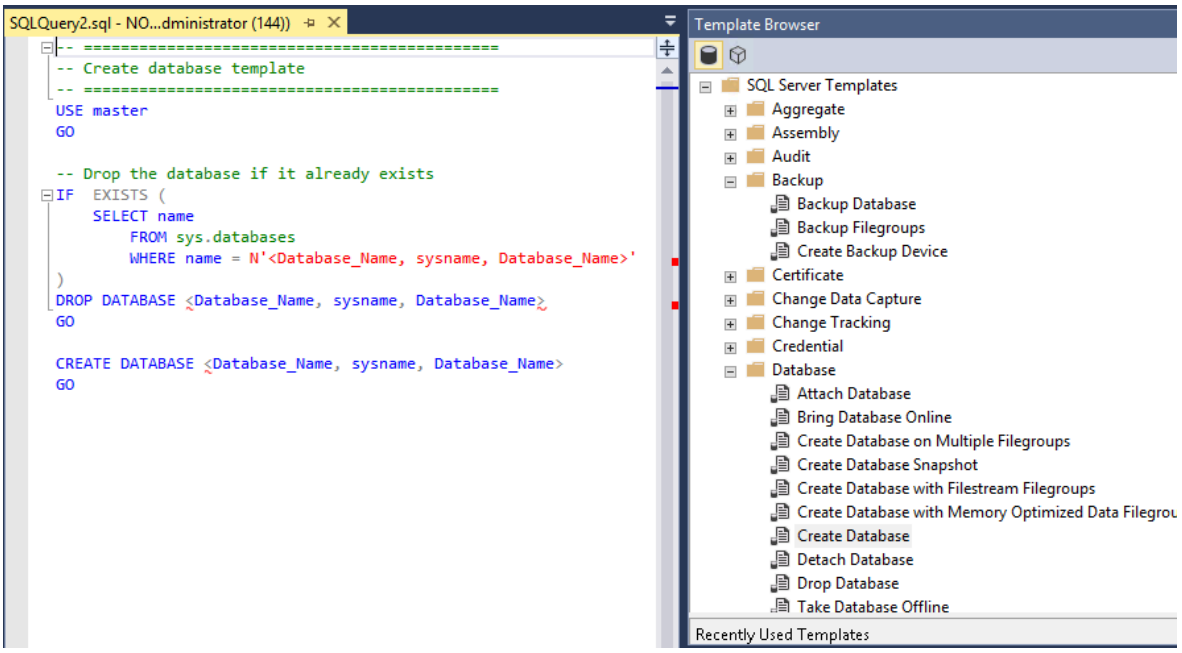
- **Query Window** (Ctrl+N): After you select **New Query**, enter your Transact-SQL (T-SQL) queries in this window. The results of your queries also appear in the Results pane.



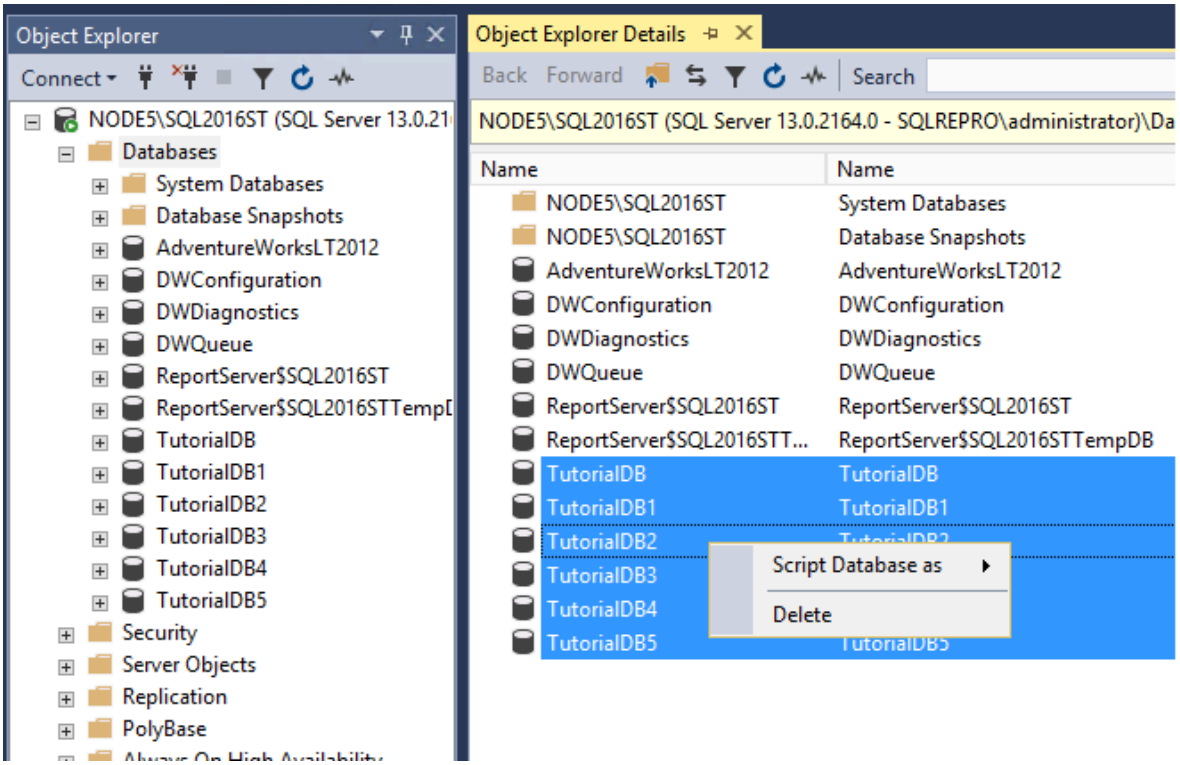
- **Current connection properties** (F4): You can see the Current connection properties pane when the Query Window is open. The view displays properties for the connection. For example, it shows the Server name, the start time for the query last executed, the number of rows returned, and other connection details.



- **Template Browser** (Ctrl+Alt+T): The Template Browser has an extensive set of prebuilt T-SQL templates. You can use these templates to perform various functions, such as creating or backing up a database, or creating an index.



- **Object Explorer Details** (F7): The details pane provides more granular information compared to Object Explorer. You can use the Object Explorer Details pane to manipulate multiple objects at the same time. For example, you can select multiple databases and then script them out simultaneously.

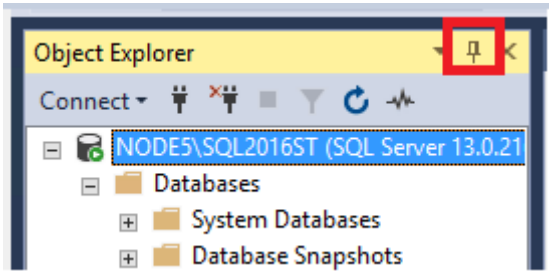


Change the environment layout

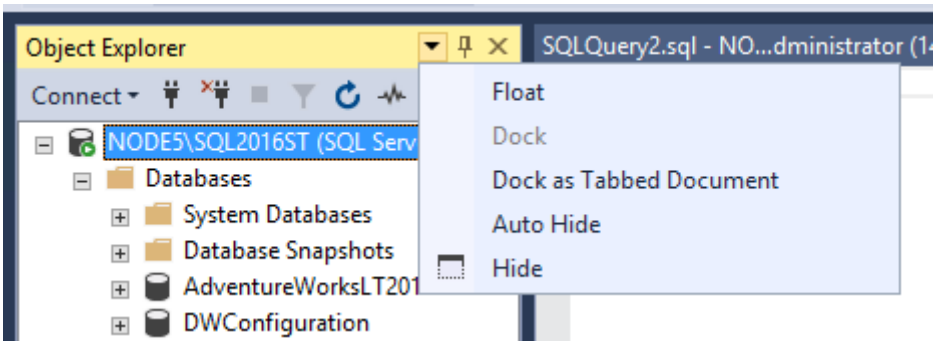
This section describes how to change the environment layout, such as how to move various windows.

- To move a window, select and hold the title, and then drag the window.

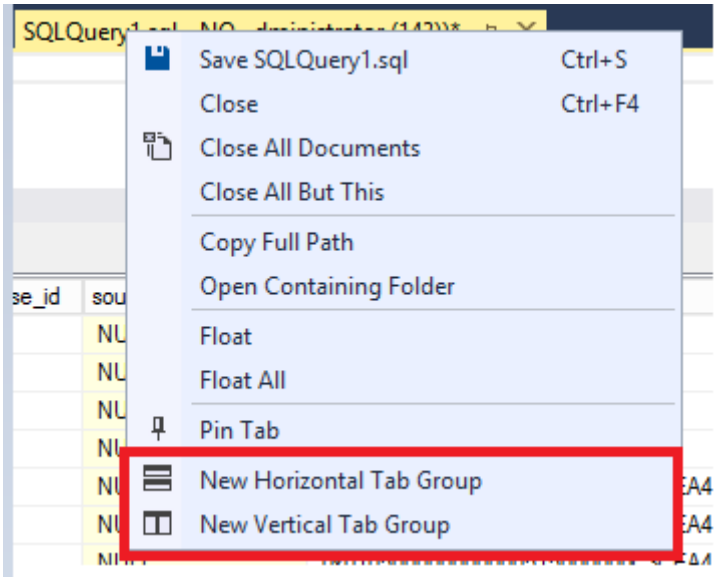
- To pin or unpin a window, select the pushpin icon in the title bar:



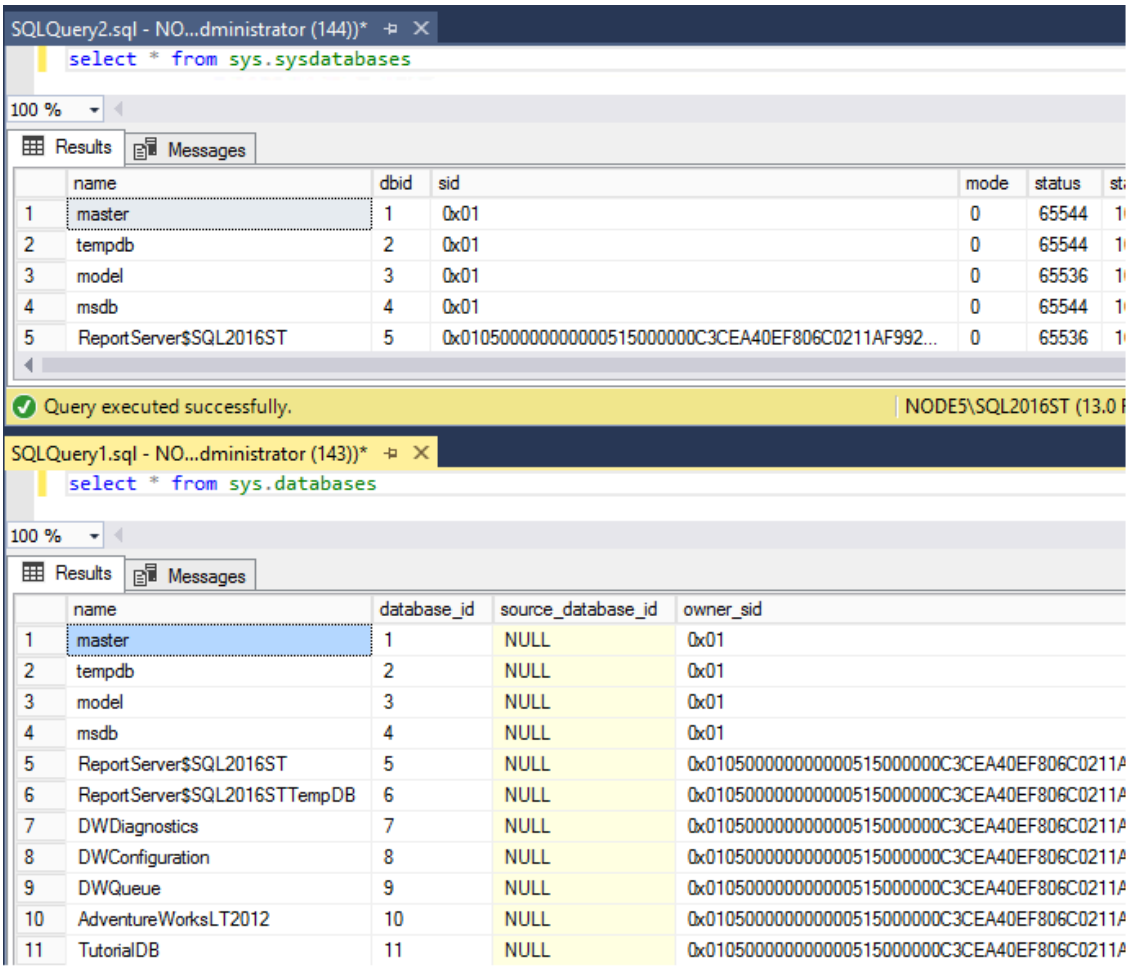
- Each window component has a drop-down menu that you can use to manipulate the window in various ways:



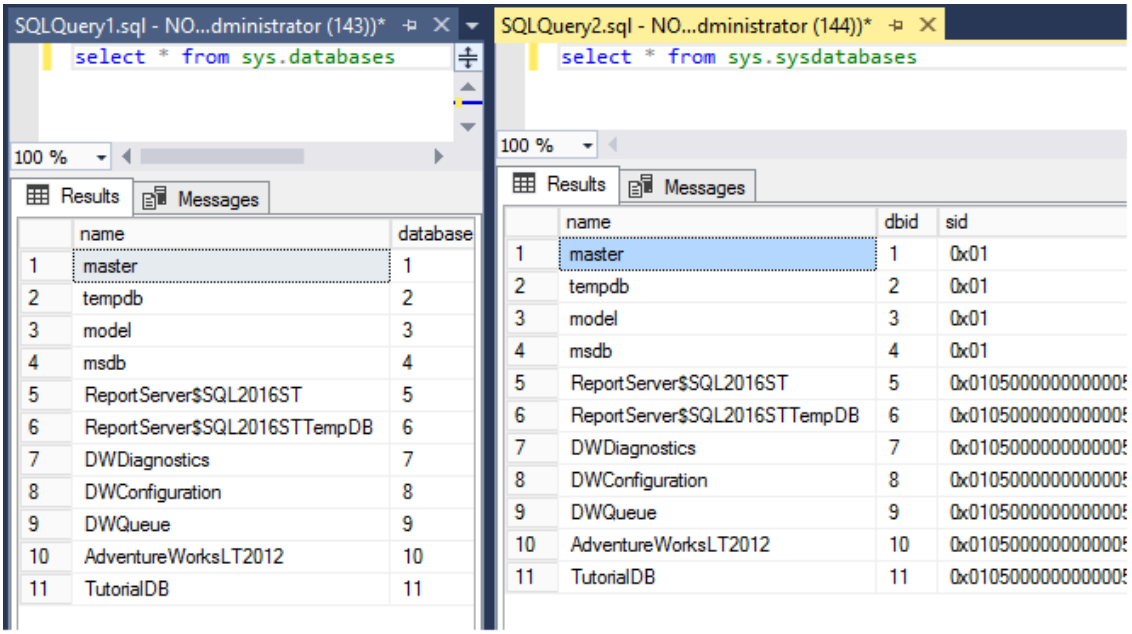
- When two or more query windows are open, the windows can be separated into multiple tab groups so that all query windows are visible. Tab groups can be vertical or horizontal. To create a tab group, right-click the title of the query, and then select the tabbed option that you want:



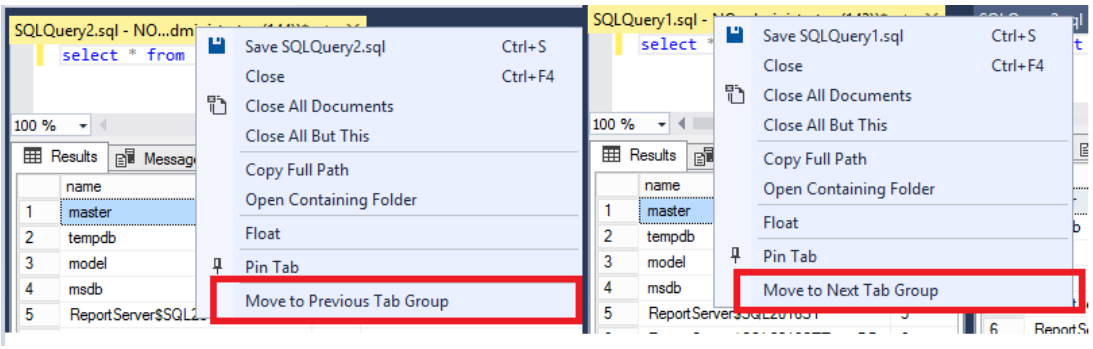
  - Horizontal Tab Group:



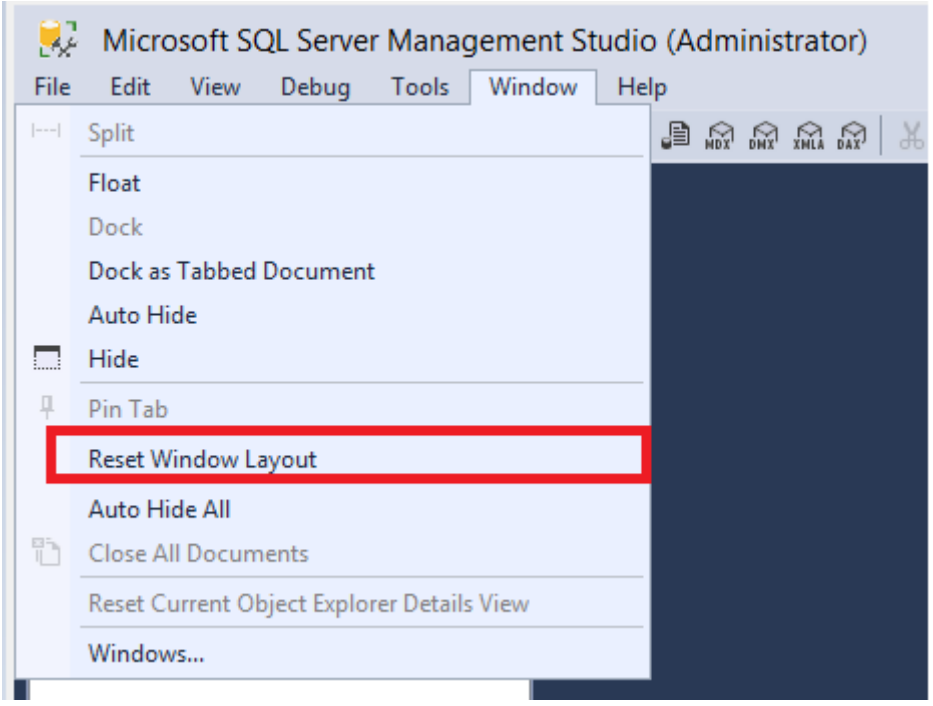
- Vertical Tab Group:



- To merge a query window into another tabs, right-click the query title, and then select **Move to Previous Tab Group** or **Move to Next Tab Group**:



- To restore the default environment layout, in the **Window** menu, select **Reset Window Layout**:



Maximize Query Editor

To maximize Query Editor to full-screen mode:

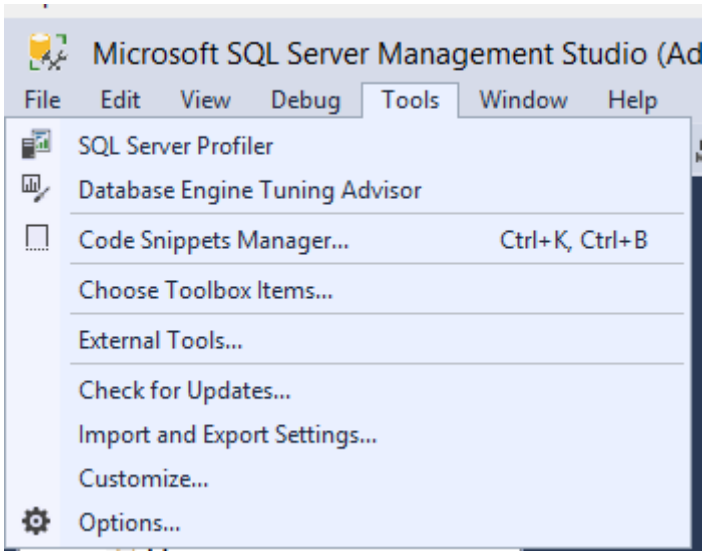
1. Click anywhere in the Query Editor window.

2. Press Shift+Alt+Enter to toggle between full-screen mode and regular mode.

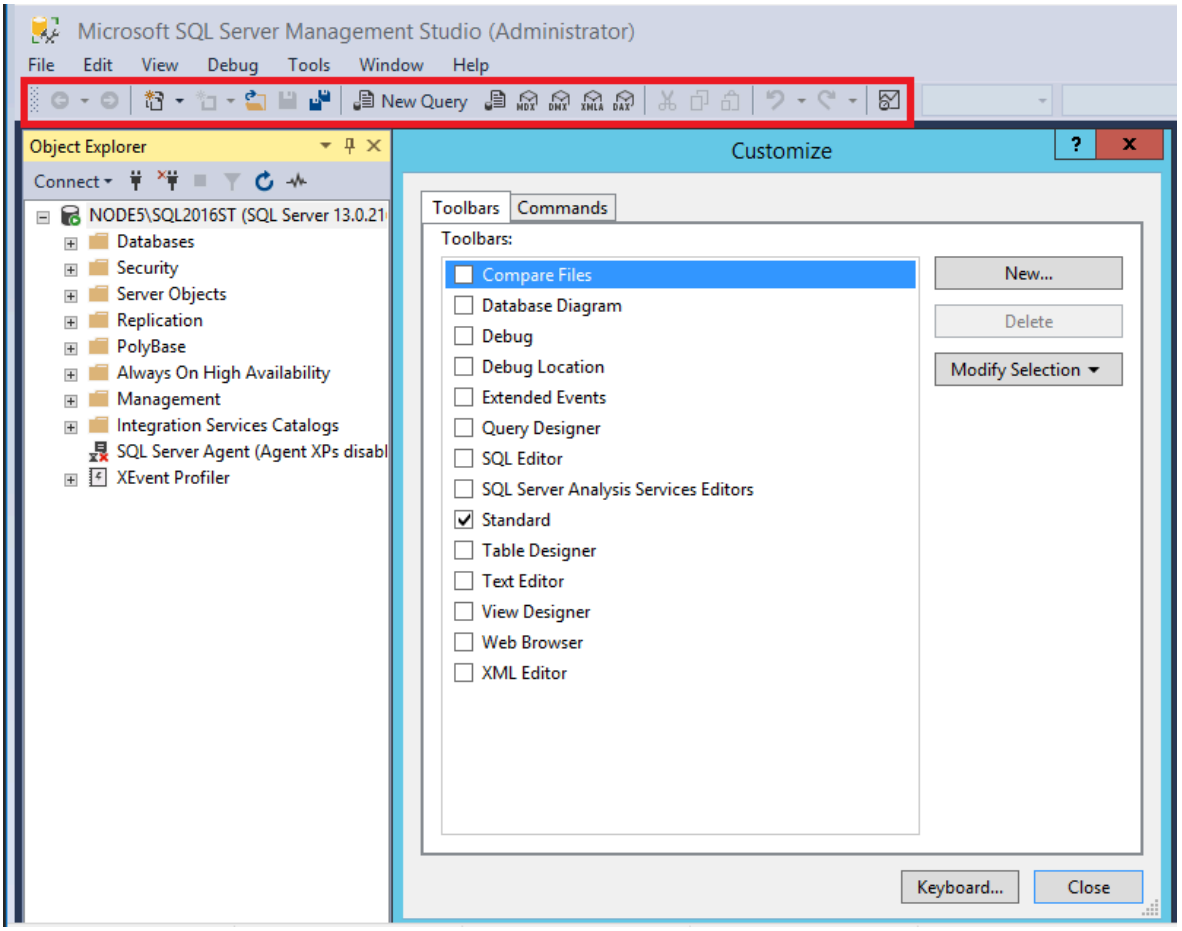
This keyboard shortcut works with any document window.

Change basic settings

This section describes how to modify some basic settings in SSMS from the **Tools** menu.



- To modify a toolbar, select **Tools** > **Customize**, or right-click on the toolbar and select **Customize...**:

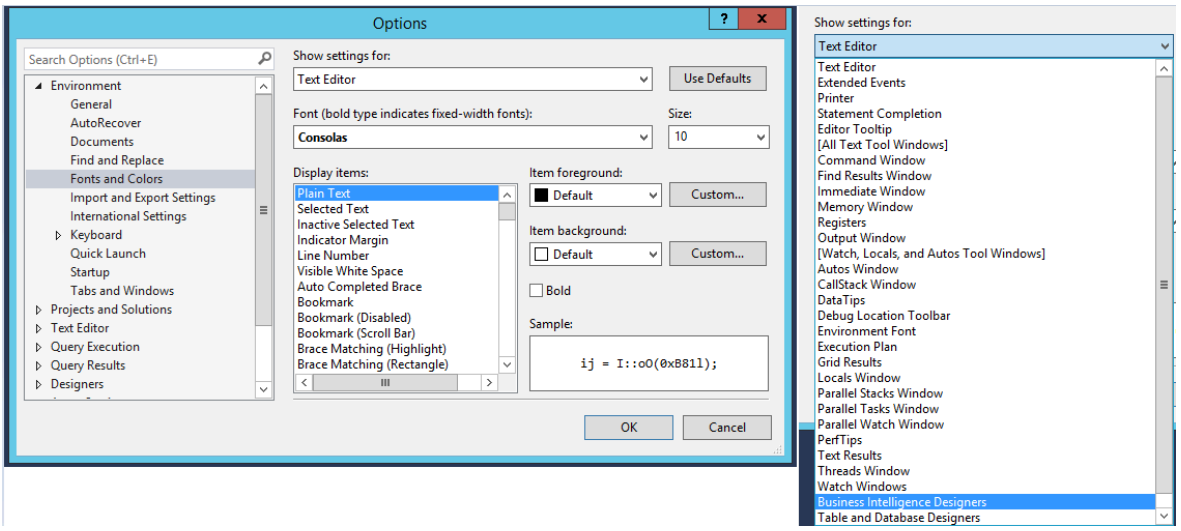


  - Use the **Toolbars** pane to enable or disable the toolbars displayed under the menu bar.

  - Use the **Commands** pane to customize the icons displayed for a toolbar.

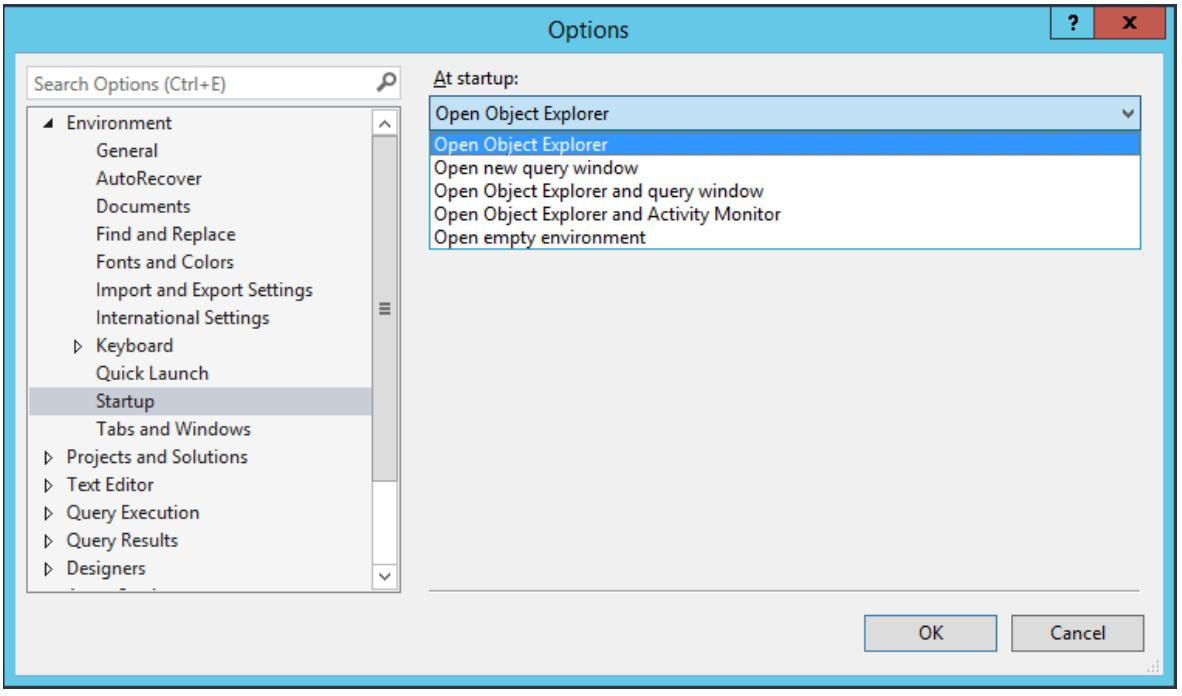
Change the font

- To change the font for a window, select **Tools** > **Options** > **Fonts and Colors**:



Change startup options

- The startup options determine what your workspace looks like when you first open SSMS. To change startup options, select **Tools** > **Options** > **Startup**:



Next steps

The best way to get acquainted with SSMS is through hands-on practice. These *tutorial* and *how-to* articles help you with various features available within SSMS. These articles teach you how to manage the components of SSMS and how to find the features that you use regularly.

- Connect to and query an instance
- Scripting
- Using Templates in SSMS

- Import and Export settings
- Tips and tricks for SSMS

Feedback

Was this page helpful? Yes No

Import and export settings for SQL Server Management Studio (SSMS)

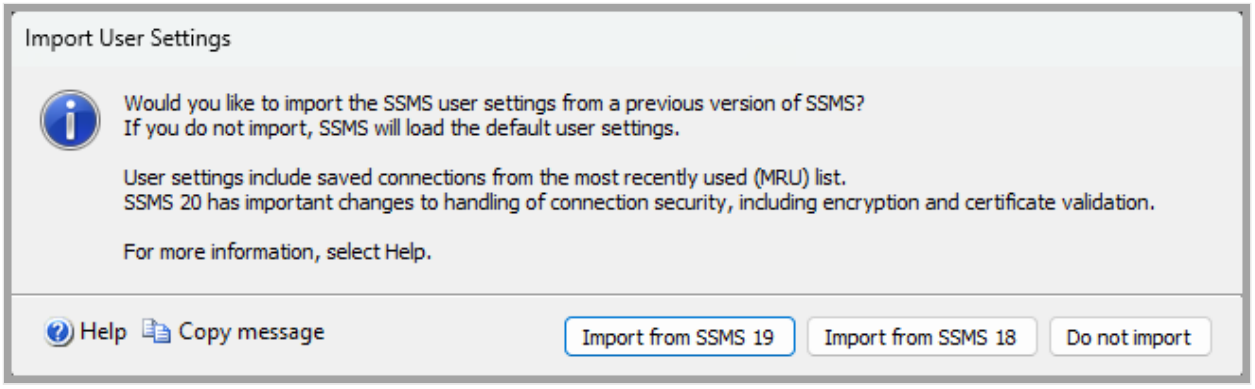
Article • 02/13/2025

This tutorial describes how to import settings from another installation of SQL Server Management Studio (SSMS), and how to export settings from SSMS. In this article, you learn how to:

- ✓ Import settings
- ✓ Export settings
- ✓ Reset all settings

Import user settings

The first time you open SSMS after installing a major release you are prompted to import user settings:



The **Import User Settings** dialog only appears on the first launch of a newly installed version. For example, if you have SSMS 19.3 installed on your machine and install SSMS 20.2, the dialog appears. If you have SSMS 20.1 installed and you install SSMS 20.2, the import dialog doesn't appear.

For a new installation of SSMS 20, you can import settings from SSMS 18 or SSMS 19, or not import any settings.

Choosing to import settings in this dialog only imports connection settings. After the import is complete, open the connection dialog. Within the **Server name** drop-down, you see the list of servers that exists in the version of SSMS from which you imported settings. The connection settings for each server, such as **Authentication** and **User name**, are also imported.

Import environment settings

1. To import environment settings from another installation of SSMS, select **Tools** > **Import and Export Settings...**.
2. In the welcome dialog, select **Import selected environment settings**, then select **Next>**.
3. If there are changes to environment settings that you want to save, select **Yes, save my current settings**, and enter a filename and location for the settings file to be saved. Otherwise, select **No, just import new settings, overwriting my current settings**. Select **Next>**.
4. Choose the settings file that you want to import, and select **Next>**.
5. Choose the settings to import, then select **Finish**.

Export environment settings

1. To export environment settings from an existing installation of SSMS, select **Tools** > **Import and Export Settings...**.
2. In the welcome dialog, select **Export selected environment settings**, then select **Next>**.
3. Choose the settings to export, then select **Next>**.
4. Entire a filename and location for the settings file to be saved, then select **Finish**.

Reset settings to the default

1. To restore default settings, select **Tools** > **Import and Export Settings...**.
2. In the welcome dialog, select **Reset all settings**, then **Next>**.
3. If there are changes to environment settings that you want to save, select **Yes, save my current settings** and enter a filename and location for the settings file to be saved. Otherwise, select **No, just import new settings, overwriting my current settings**. Select **Next>**.
4. Choose collection of settings to reset to, then select **Finish**.

Next steps

The best way to get acquainted with SSMS is through hands-on practice. These *tutorial* and *how-to* articles help you with various features available within SSMS. These articles teach you how to manage the components of SSMS and how to find the features that you use regularly.

- Connect to and query an instance

- SSMS components and configuration
- Scripting
- Using Templates in SSMS
- Tips and tricks for SSMS

Feedback

Was this page helpful? Yes No

Tips and tricks for using SQL Server Management Studio (SSMS)

Article • 02/13/2025

This article includes some tips and tricks for using SQL Server Management Studio (SSMS). This article shows you how to:

- ✓ Comment/uncomment your Transact-SQL (T-SQL) text
- ✓ Indent your text
- ✓ Filter objects in Object Explorer
- ✓ Access your SQL Server error log
- ✓ Find the name of your SQL Server instance

Prerequisites

To test the steps provided in this article, you need SQL Server Management Studio, access to an instance of SQL Server and an AdventureWorks database.

- Install SQL Server Management Studio.
- Install SQL Server Developer Edition.
- Download an AdventureWorks sample database. To learn how to restore a database in SSMS, see Restoring a database.

Comment/uncomment your T-SQL code

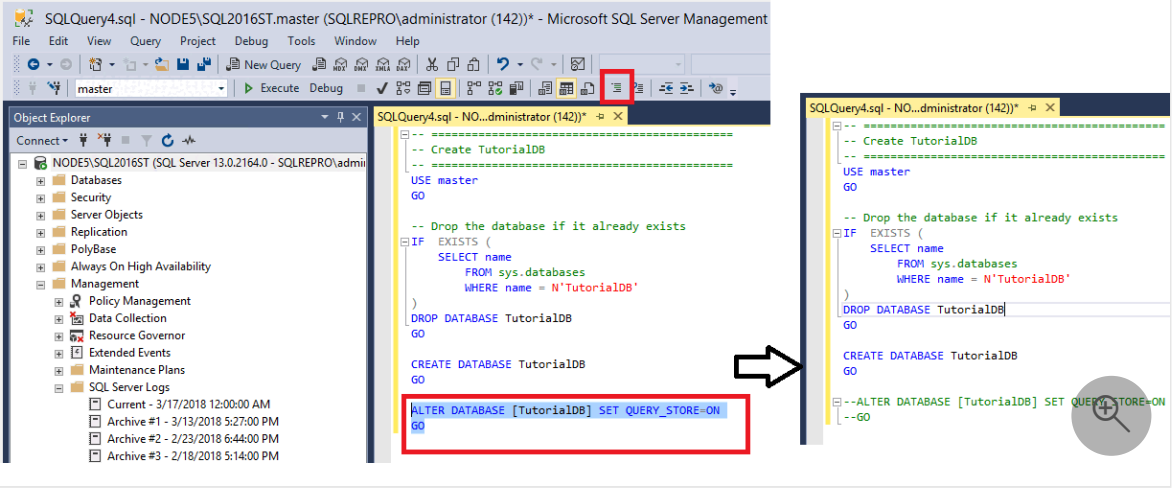
You can comment and uncomment portions of your text by using the **Comment** button on the toolbar. Text that is commented out isn't executed.

1. Open SQL Server Management Studio.
2. Connect to your SQL Server instance.
3. Open a New Query window.
4. Paste the following Transact-SQL code in your text window.

```SQL
USE master;
GO

--Drop the database if it already exists
```

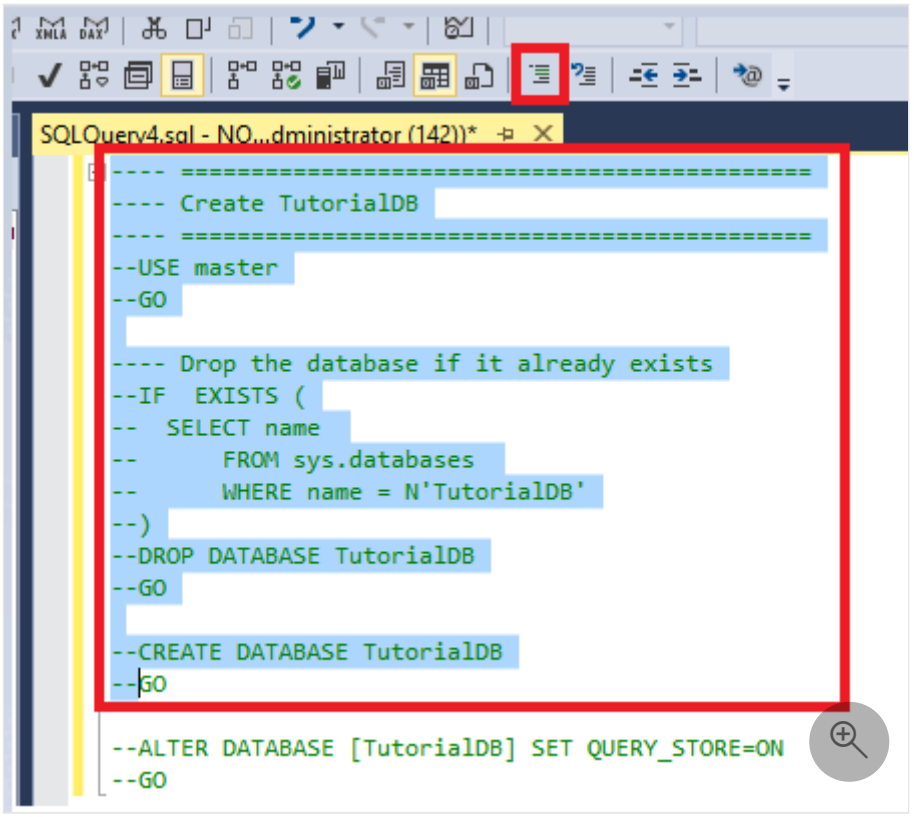
```
IF EXISTS (SELECT name
           FROM sys.databases
           WHERE name = N'TutorialDB')
    DROP DATABASE TutorialDB;
GO

CREATE DATABASE TutorialDB;
GO

ALTER DATABASE [TutorialDB]
    SET QUERY_STORE = ON;
GO
```

5. Highlight the **Alter Database** portion of the text, and then select the **Comment** button on the toolbar:



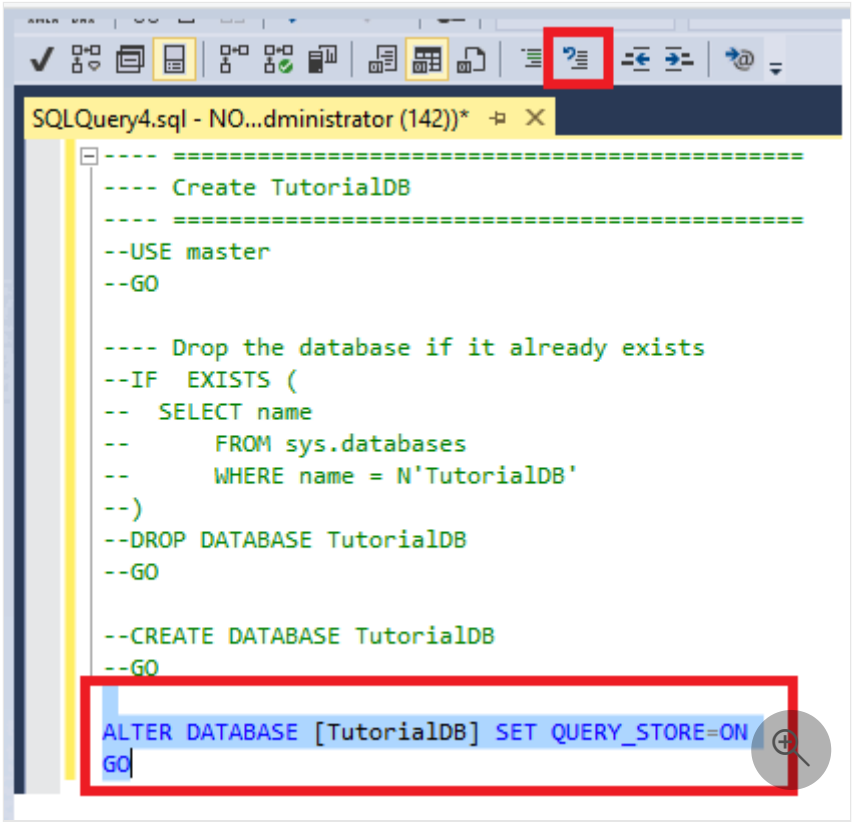
6. Select **Execute** to run the uncommented portion of the text.

7. Highlight everything except for the **Alter Database** command, and then select the **Comment** button:



> **Note**
>
> The keyboard shortcut to comment text is **CTRL + K, CTRL + C**.

8. Highlight the **Alter Database** portion of the text, and then select the **Uncomment** button to uncomment it:



> ⓘ **Note**
>
> The keyboard shortcut to uncomment text is **CTRL + K, CTRL + U**.

9. Select **Execute** to run the uncommented portion of the text.

Indent your text

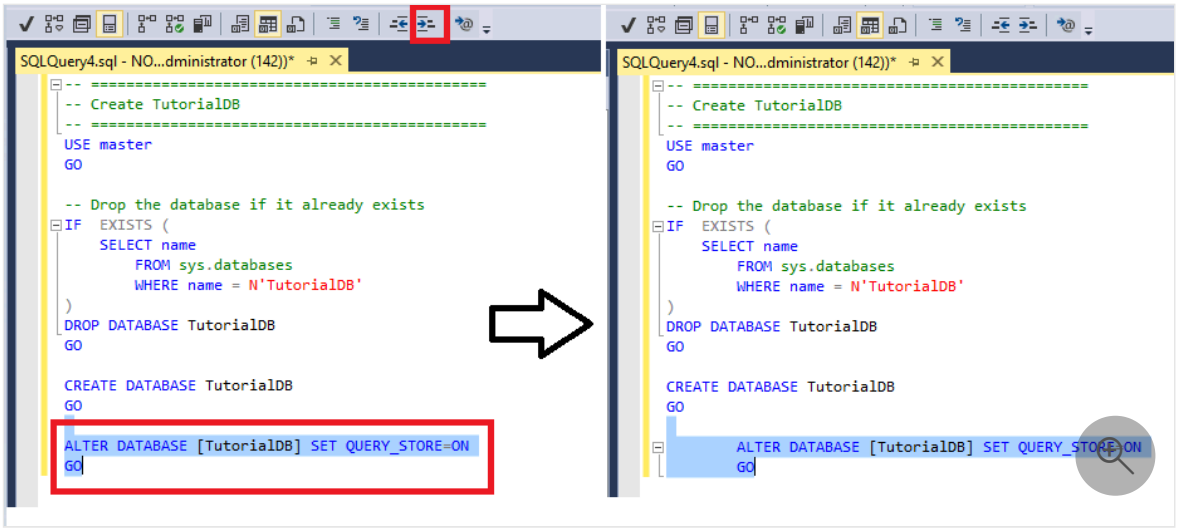
You can use the indentation buttons on the toolbar to increase or decrease the indent of your text.

1. Open a New Query window.

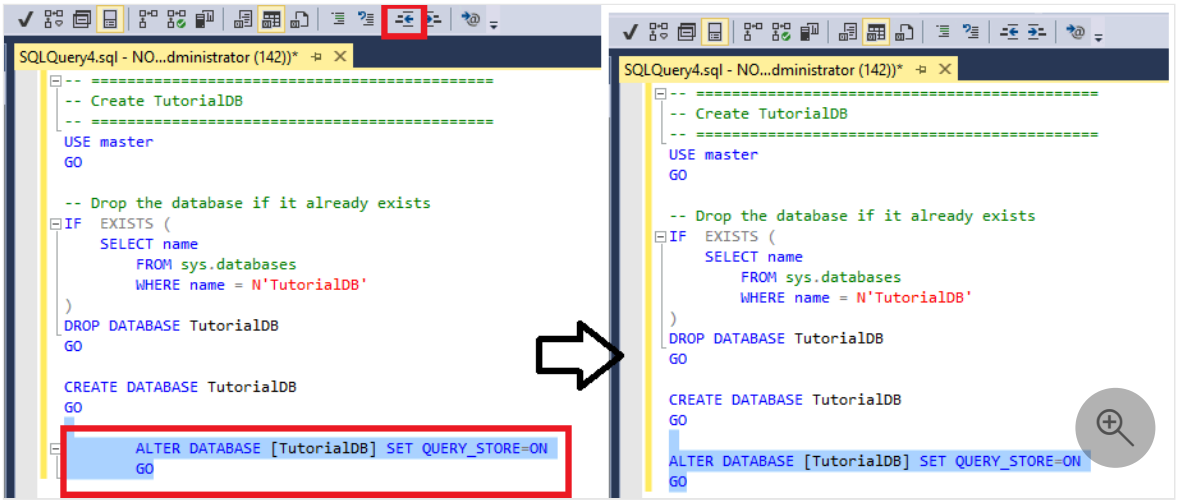
2. Paste the following Transact-SQL code in your text window:

```SQL
USE master;
GO
--Drop the database if it already exists
IF EXISTS (SELECT name
           FROM sys.databases
           WHERE name = N'TutorialDB')
    DROP DATABASE TutorialDB;
GO

CREATE DATABASE TutorialDB;
GO

ALTER DATABASE [TutorialDB]
    SET QUERY_STORE = ON;
GO
```

3. Highlight the **Alter Database** portion of the text, and then select the **Increase Indent** button on the toolbar to move the highlighted text forward:



4. Highlight the **Alter Database** portion of the text again, and then select the **Decrease Indent** button to move the highlighted text back.



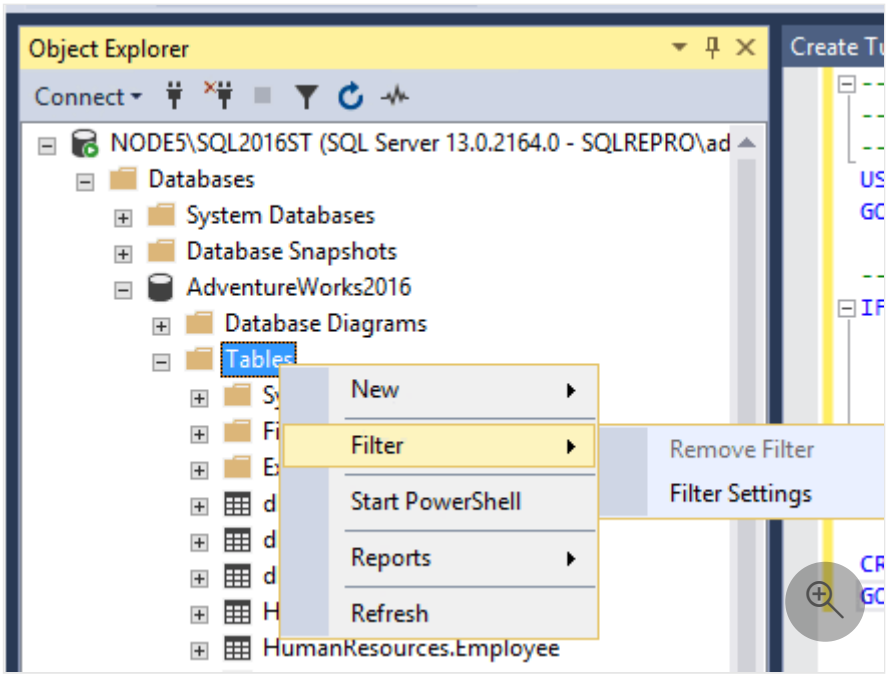
Filter objects in Object Explorer

In databases that have many objects, you can use filtering to search for specific tables, views, etc. This section describes how to filter tables, but you can use the following steps in any other nodes in Object Explorer:

1. Connect to your SQL Server instance.

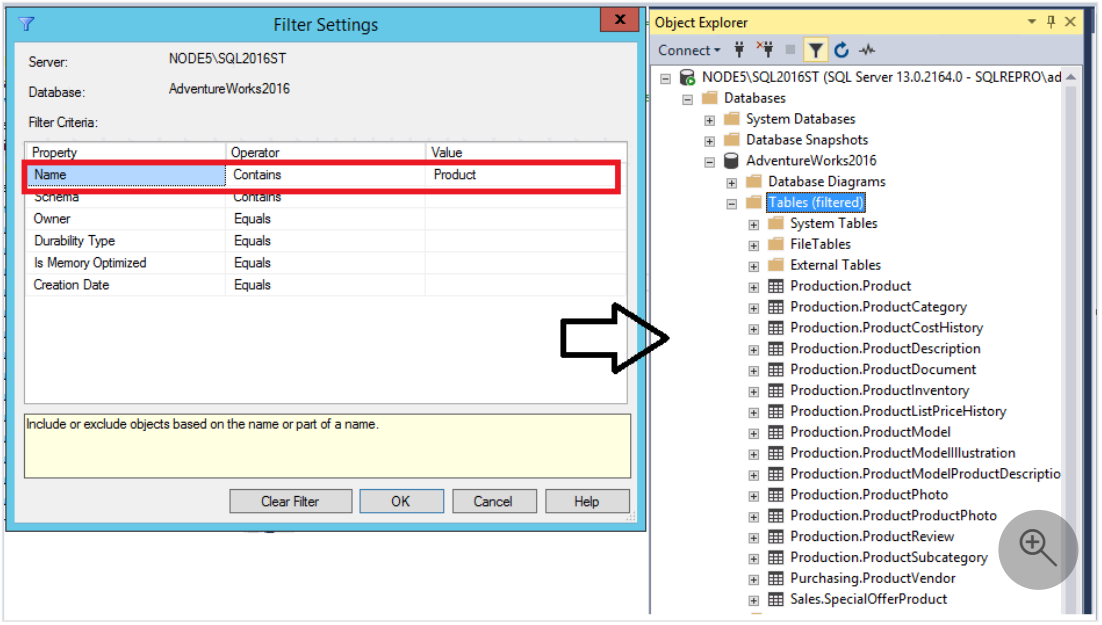
2. Expand **Databases** > **AdventureWorks** > **Tables**. All the tables in the database appear.

3. Right-click **Tables**, and then select **Filter** > **Filter Settings**:

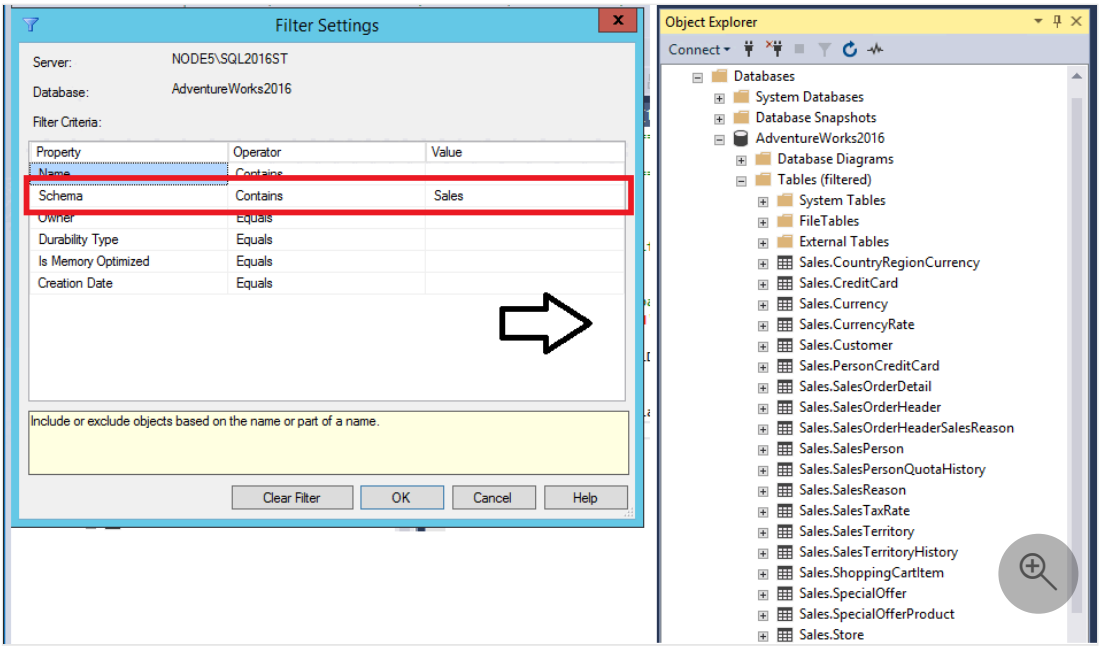


4. In the **Filter Settings** window, you can modify some of the following filter settings:

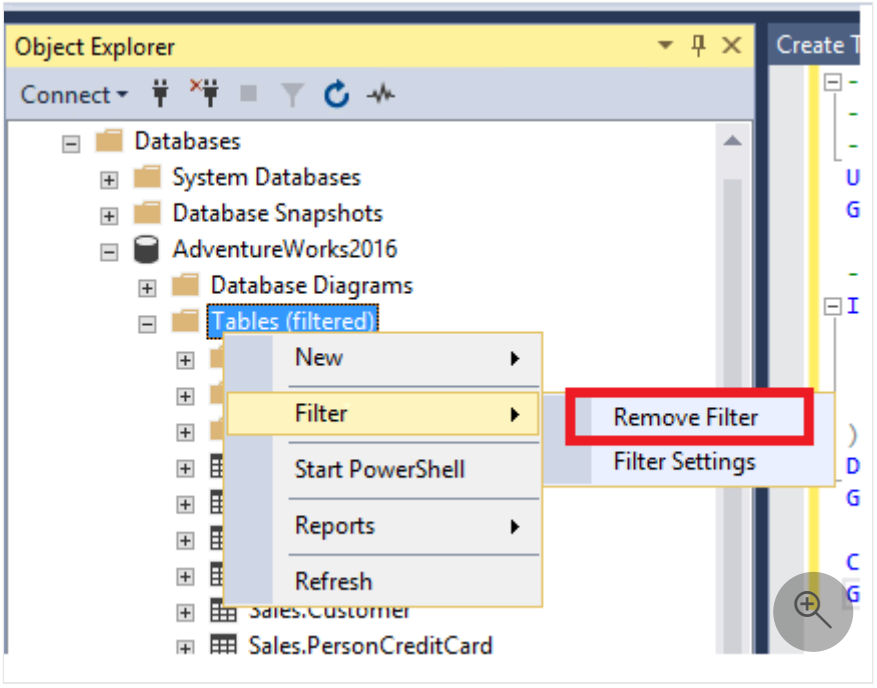
5.
   - Filter by name:



   - Filter by schema:



6. To clear the filter, right-click **Tables**, and then select **Remove Filter**.



Access your SQL Server error log

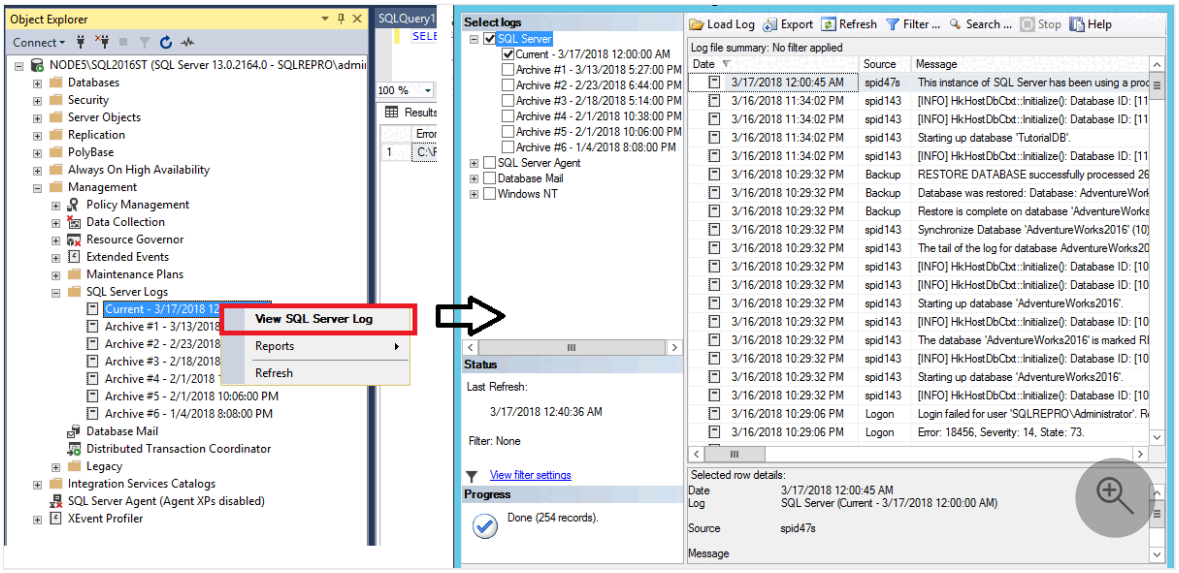
The error log is a file that contains details about things that occur in your SQL Server instance. You can browse and query the error login SSMS. The error log is a .log file that exists in your file system.

Open the error log in SSMS

1. Connect to your SQL Server instance.

2. Expand **Management** > **SQL Server Logs**.

3. Right-click the **Current** error log, and then select **View SQL Server Log**:



Query the error log in SSMS

1. Connect to your SQL Server instance.

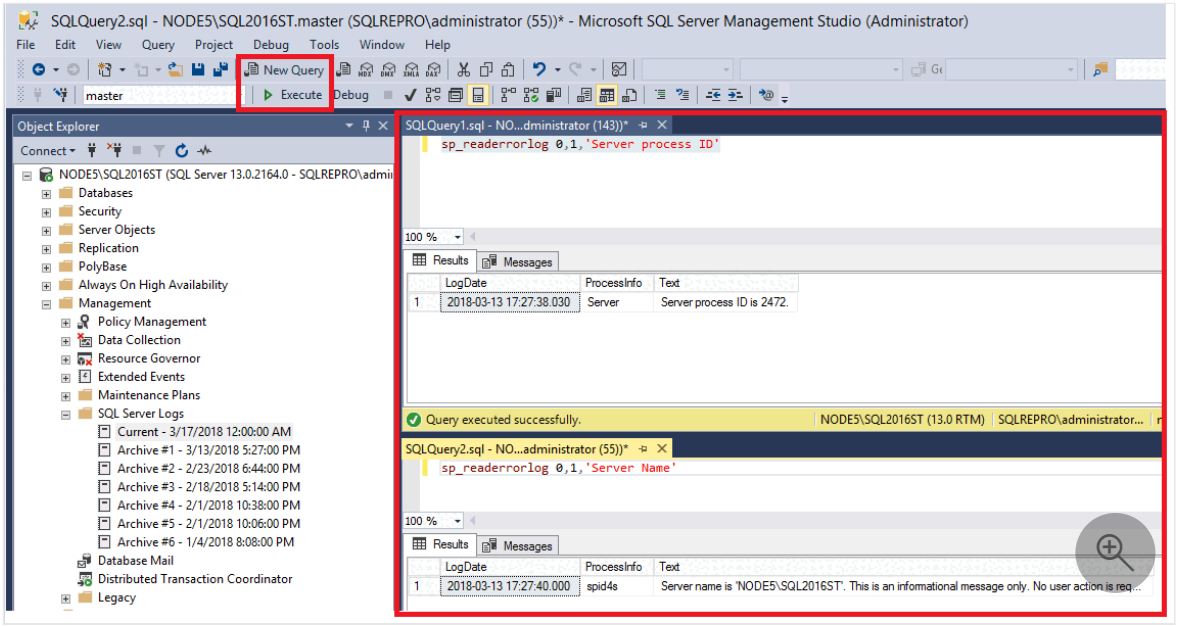
2. Open a **New Query** window.

3. Paste the following Transact-SQL code in your query window:

```
SQL

EXECUTE sp_readerrorlog 0, 1,'Server process ID'
```

4. Modify the text in the single quotes to text you want to search for.

5. Execute the query, and then review the results:

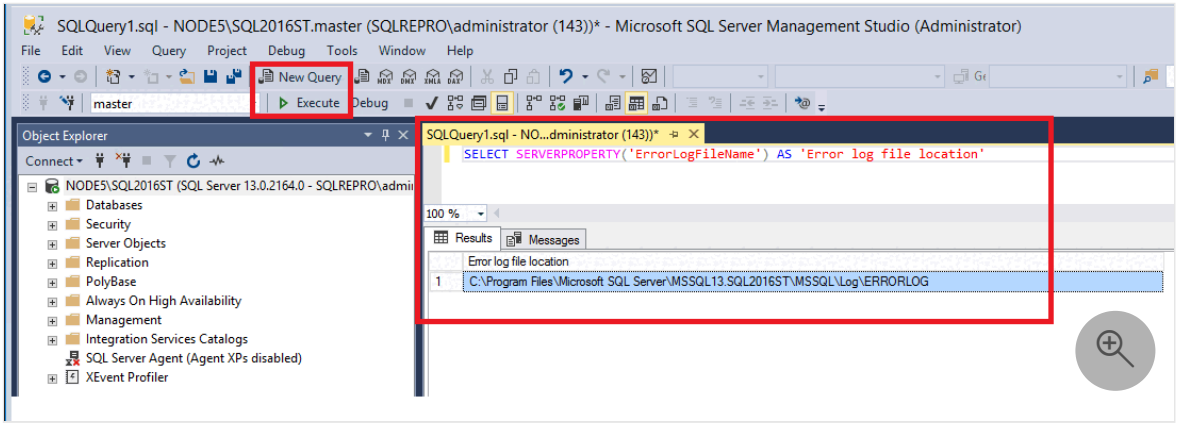


Find the error log location if you're connected to SQL Server

1. Connect to your SQL Server instance.
2. Open a **New Query** window.
3. Paste the following Transact-SQL code in your query window, and then select **Execute**:

```
SQL

SELECT SERVERPROPERTY('ErrorLogFileName') AS 'Error log file location';
```

4. The results show the location of the error log in the file system:

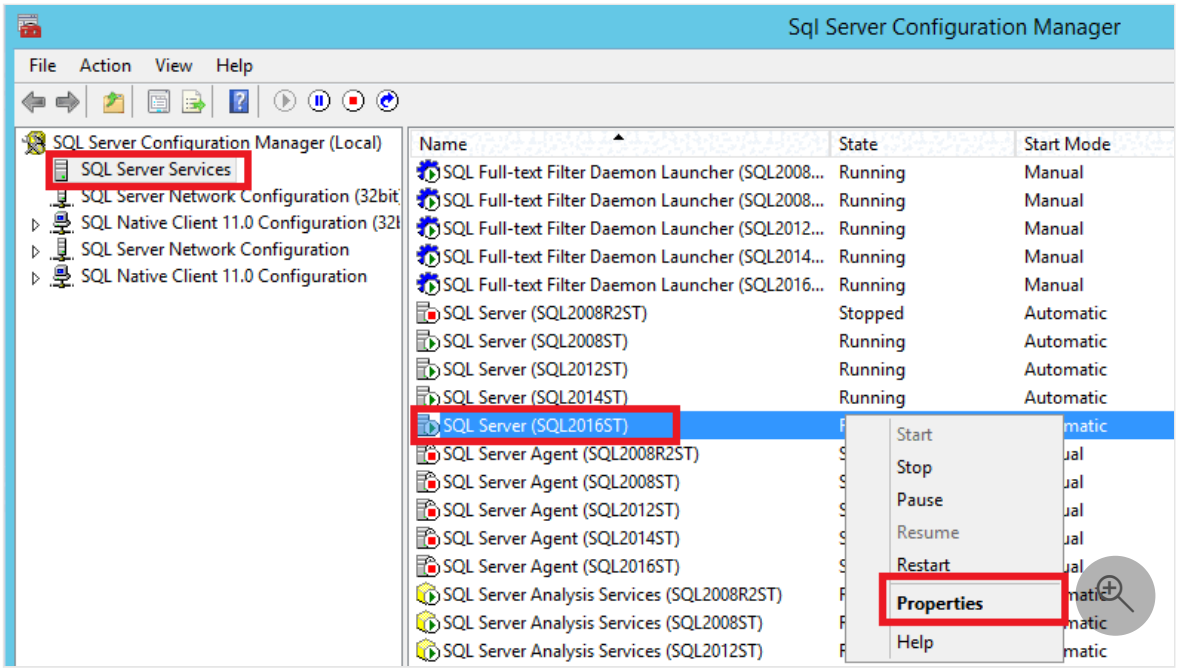


Find the error log location if you can't connect to SQL Server

The path for your SQL Server error log can vary depending on your configuration settings. The path for the error log location can be found in the SQL Server startup parameters within the SQL Server Configuration Manager.

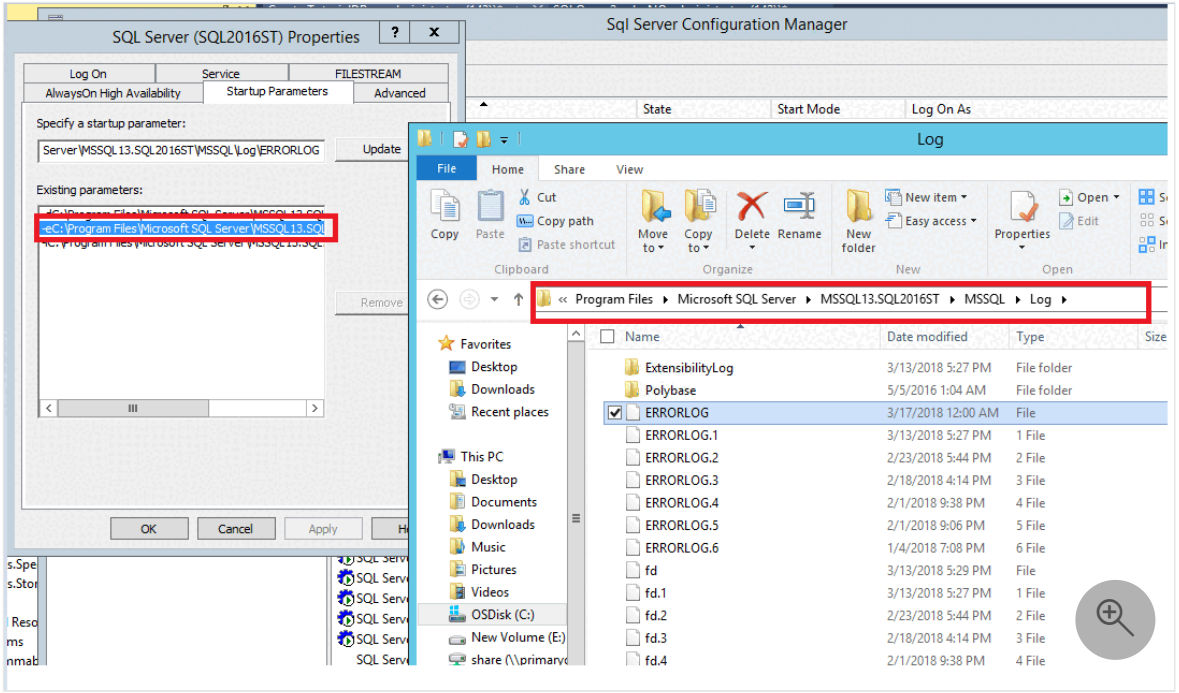
Follow these steps to locate the relevant startup parameter that identifies the location of your SQL Server error log. *Your path might vary from the path indicated in the example*.

1. Open SQL Server Configuration Manager.
2. Expand **Services**.
3. Right-click your SQL Server instance, and then select **Properties**:



4. Select the **Startup Parameters** tab.

5. In the **Existing Parameters** area, the path after `-e` is the location of the error log:



There are several error log files in this location. The file name that ends with *log is the current error log file. File names that end with numbers are previous log files. A new log is created every time the SQL Server restarts.

6. Open the errorlog.log file in your preferred text editor.

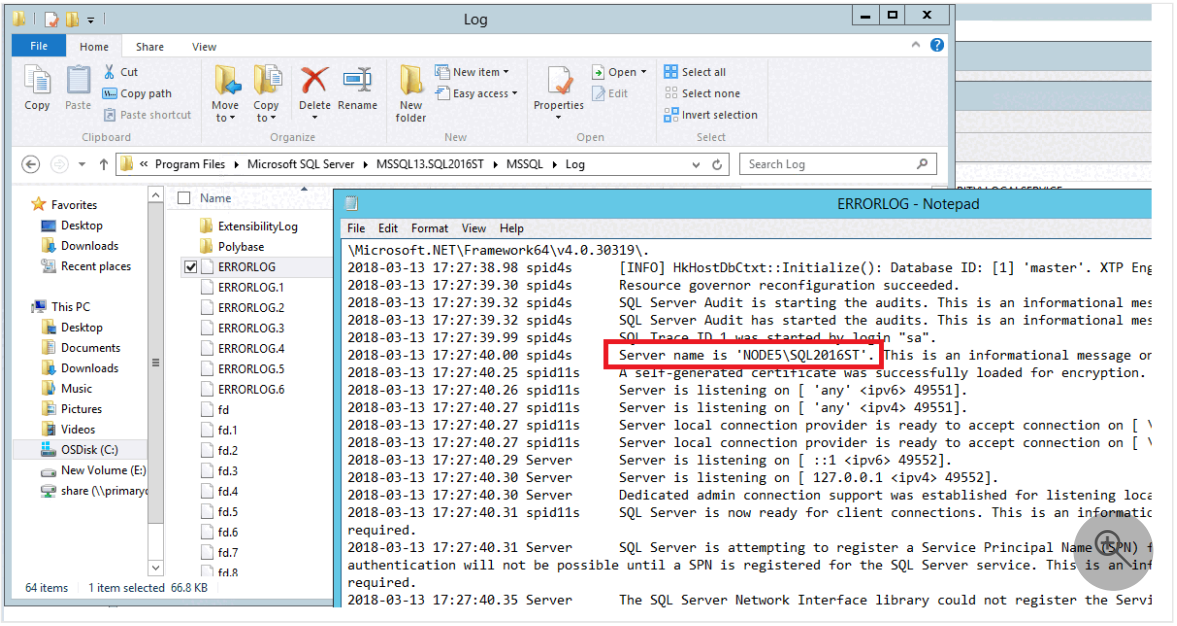
Find SQL Server instance name

You have a few options to find the name of your SQL Server instance before and after you connect to SQL Server.

Before you connect to SQL Server

1. Follow the steps to locate the SQL Server error log on disk.
2. Open the errorlog.log file in Notepad.
3. Search for the text *Server name is*.

   The text listed in the single quotes is the name of the SQL Server instance to use in the connection:

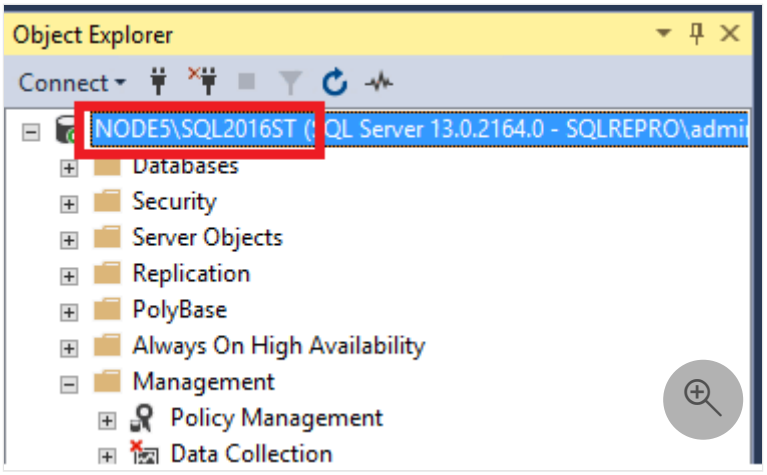


   The format of the name is `HOSTNAME\INSTANCENAME`. If you see only the host name, then you've installed the default instance and your instance name is `MSSQLSERVER`. When you connect to a default instance, the host name is all you need to enter to connect to your SQL Server. Your path might vary from the path in the sample image.

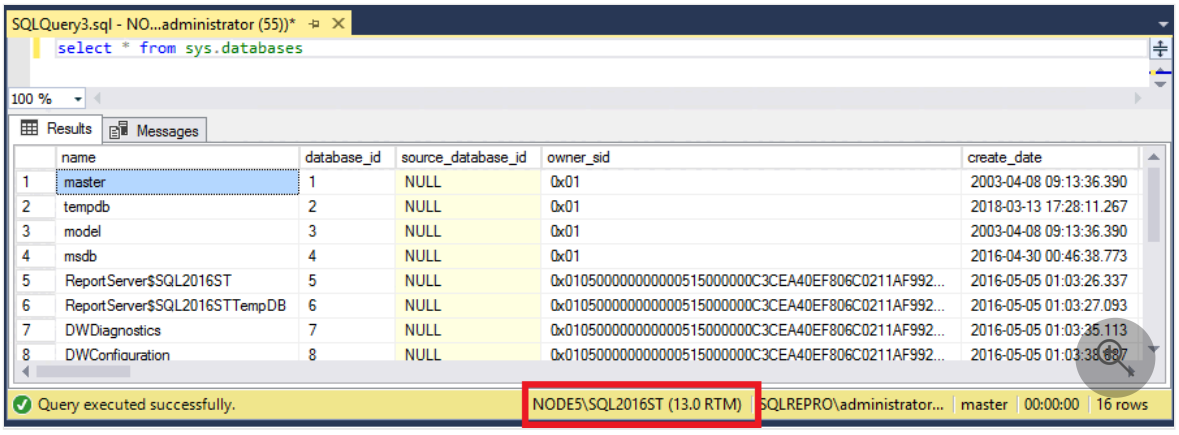
When you're connected to SQL Server

When you're connected to SQL Server, you can find the server name in three locations:

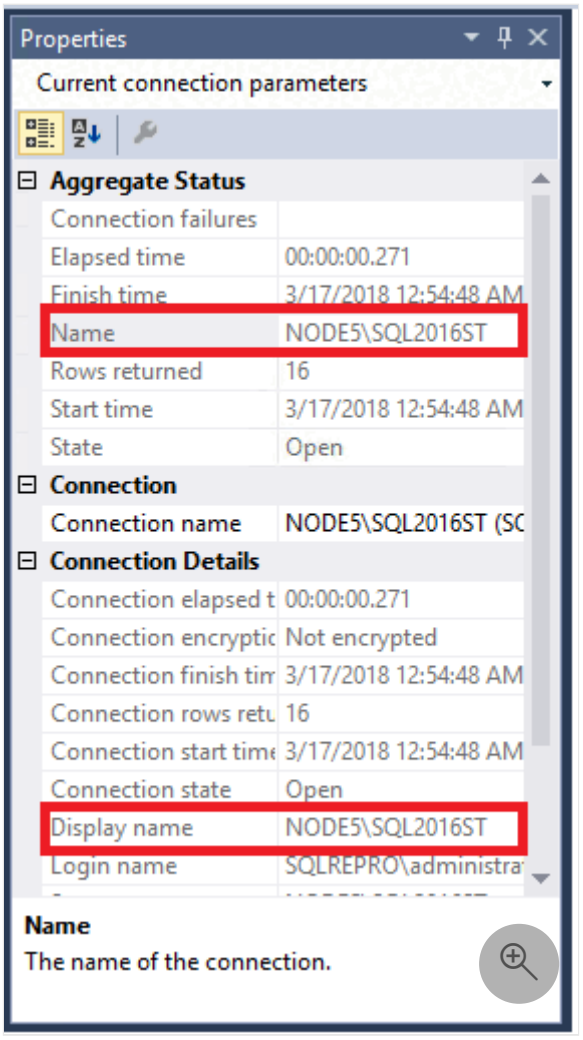
- The name of the server is listed in Object Explorer:



- The name of the server is listed in the Query window:



- The name of the server is listed in **Properties**.
    - In the **View** menu, select **Properties Window**:



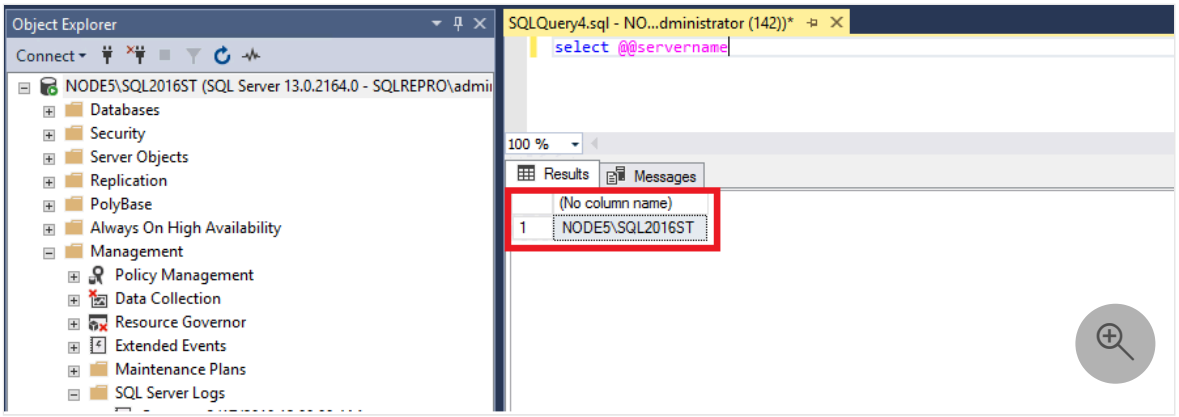
If you're connected to an alias or availability group listener

If you're connected to an alias or to an availability group listener, that information appears in **Object Explorer** and Properties. In this case, the SQL Server instance name might not be readily apparent, and must be queried:

1. Connect to your SQL Server instance.
2. Open a **New Query** window.
3. Paste the following Transact-SQL code in the window:

```
SQL

SELECT @@Servername;
```

4. View the results of the query to identify the name of the SQL Server instance you're connected to:



Related content

- Quickstart: Connect and query a SQL Server instance using SQL Server Management Studio (SSMS)
- Script objects in SQL Server Management Studio
- Use templates in SQL Server Management Studio
- SQL Server Management Studio components and configuration

Feedback

Was this page helpful? Yes No

Script objects in SQL Server Management Studio

Article • 02/13/2025

This tutorial teaches you to generate Transact-SQL (T-SQL) scripts for various objects found within SQL Server Management Studio (SSMS). In this tutorial, you find examples of how to script the following objects:

- ✓ Queries, when you perform actions within the GUI
- ✓ Databases in two different ways (Script As and Generate Script)
- ✓ Tables
- ✓ Stored procedures
- ✓ Extended events

To script any object in **Object Explorer**, right-click it and select the **Script Object As** option. This tutorial shows you the process.

Prerequisites

To complete this tutorial, you need SQL Server Management Studio, access to a server that's running SQL Server, and an AdventureWorks database.

- Install SQL Server Management Studio.
- Install SQL Server 2017 Developer Edition.
- Download AdventureWorks sample databases.

Instructions for restoring databases in SSMS are here: Restore a database.

Script queries from the GUI

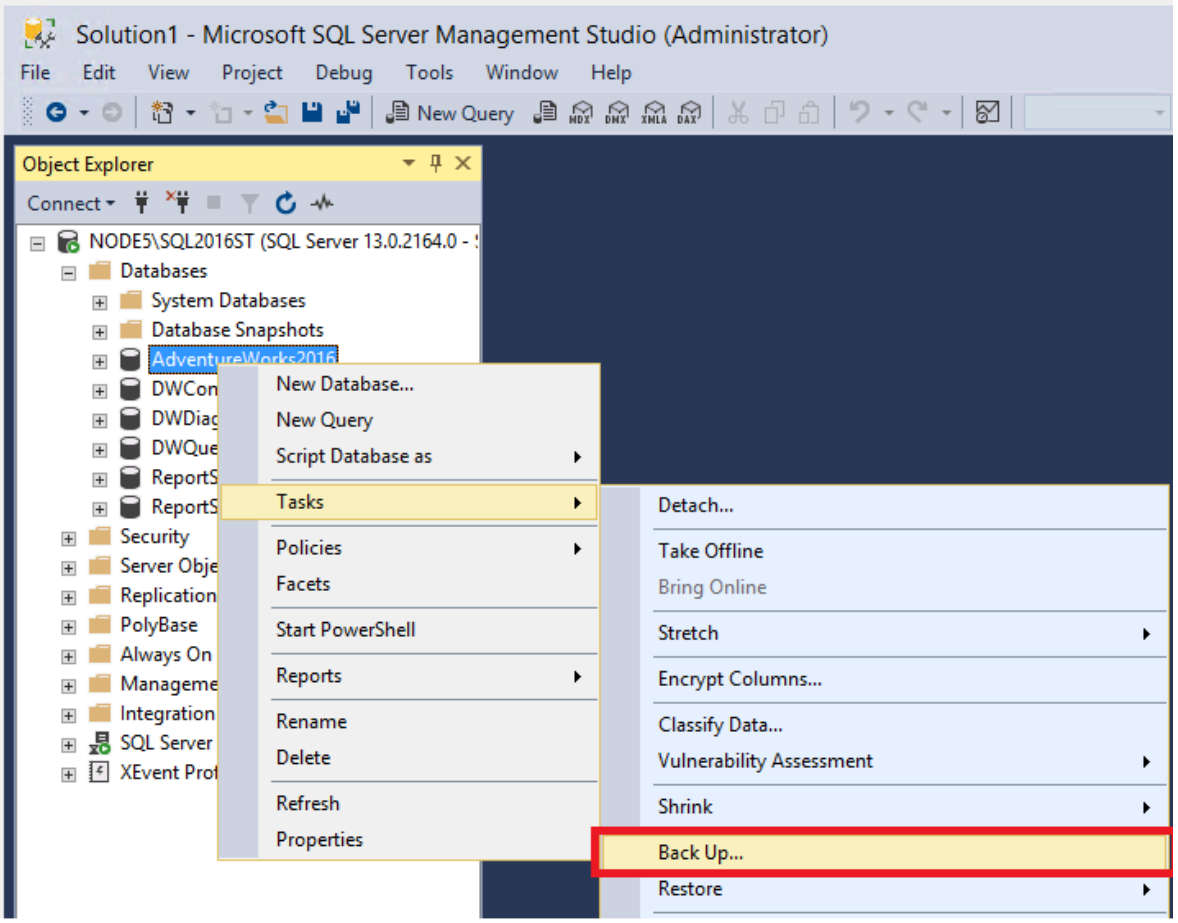
You can generate the associated T-SQL code for a task whenever you use the GUI in SSMS to complete it. The following examples show how to do so when you back up a database and when you shrink the transaction log. These same steps can be applied to any action that's completed via the GUI.

Script T-SQL when you back up a database

1. Connect to a server that's running SQL Server.

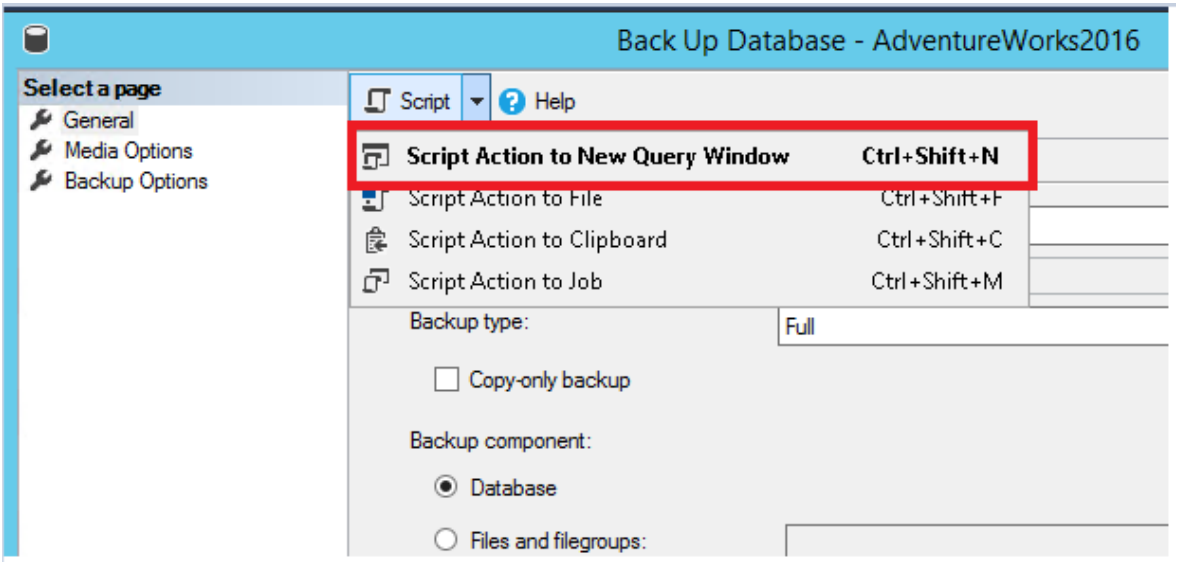
2. Expand the **Databases** node.

3. Right-click the database **AdventureWorks2022** > **Tasks** > **Back Up**:

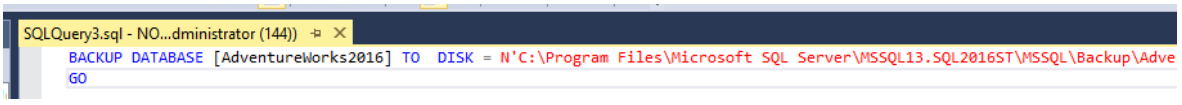


4. Configure the backup the way you want. For this tutorial, everything is left at default. However, any changes made in the window also reflect in the script.

5. Select **Script** > **Script Action to New Query Window**:



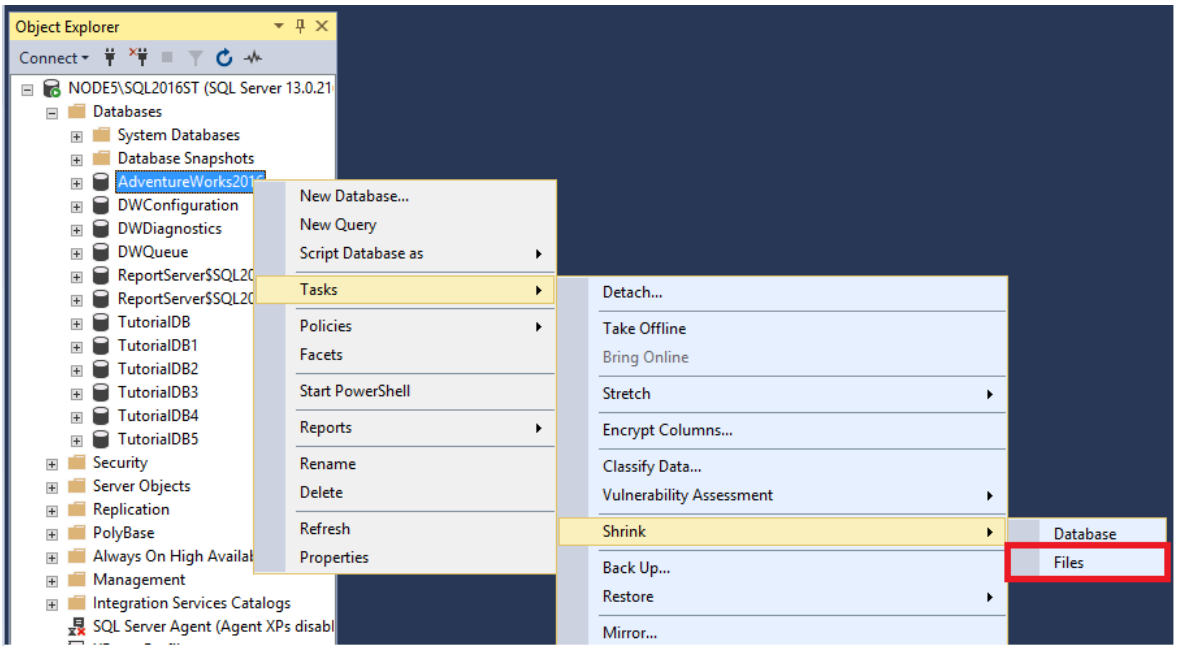
6. Review the T-SQL populated in the query window.



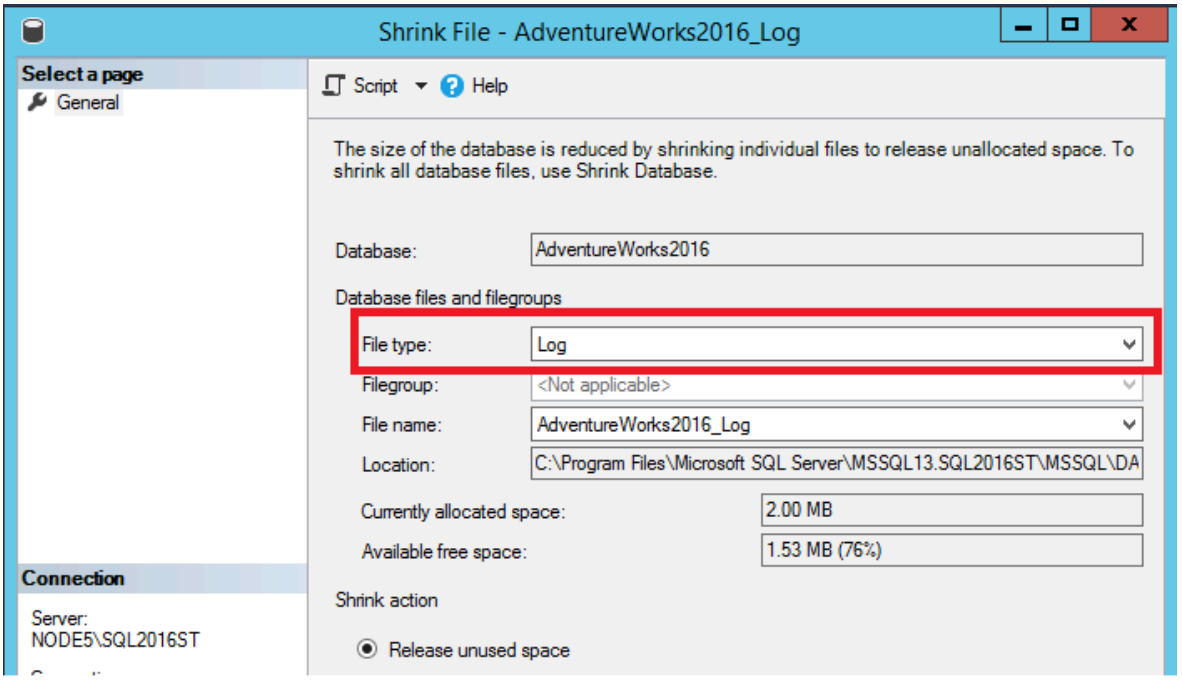
7. Select **Execute** to execute the query to back up the database via T-SQL.

Script T-SQL when you shrink the transaction log

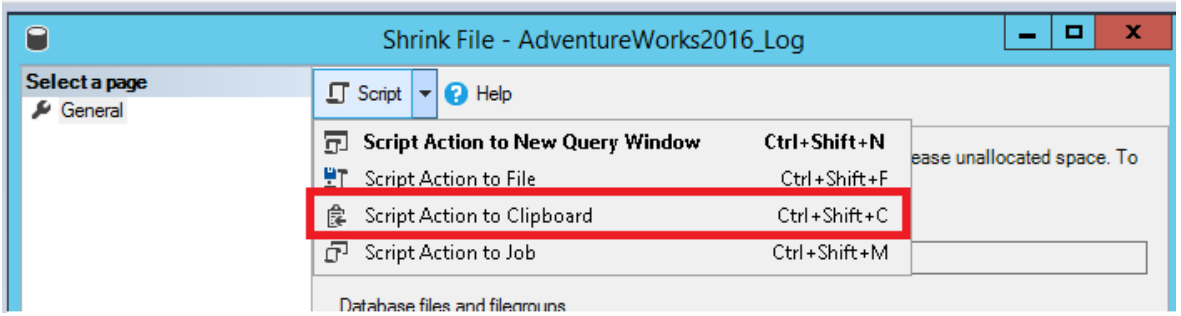
1. Right-click the database **AdventureWorks2022** > **Tasks** > **Shrink** > **Files**:



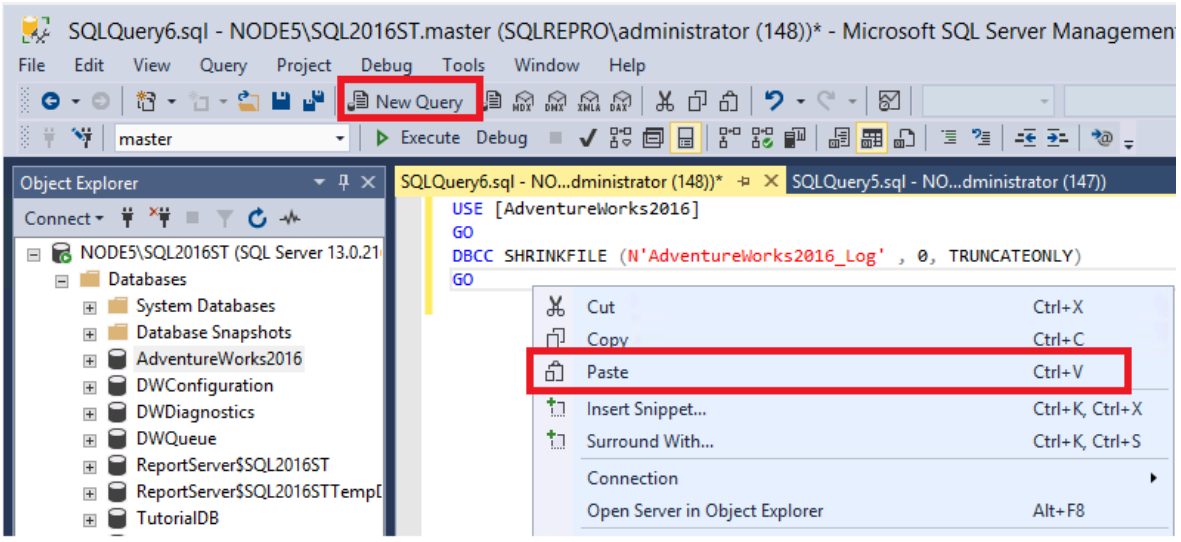
2. Select **Log** from the **File type** drop-down list box:



3. Select **Script** and **Script Action to Clipboard**:



4. Open a **New Query** window and paste. (Right-click in the window. Then select **Paste**.)



5. Select **Execute** to execute the query and shrink the transaction log.

Script databases

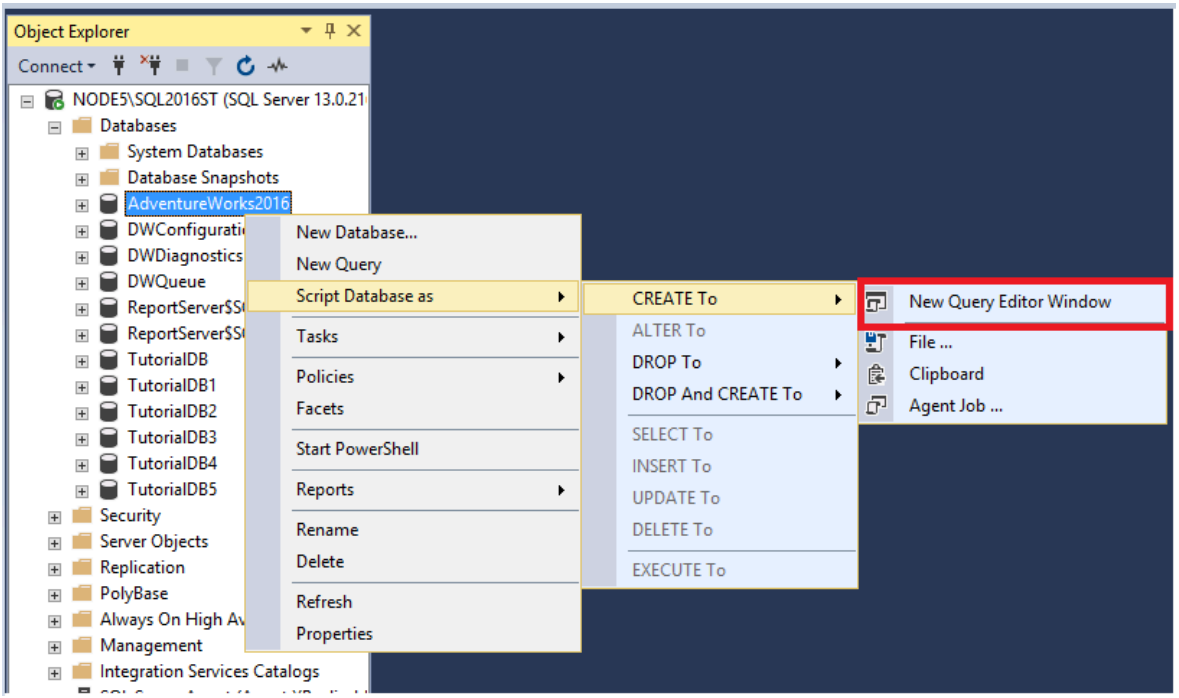
The following section teaches you to script out the database by using the **Script As** and **Generate Scripts** options. The **Script As** option re-creates the database and its configuration options. You can script both the schema and the data by using the **Generate Scripts** option. In this section, you create two new databases. You use the **Script As** option to create *AdventureWorks2022a*. You use the **Generate Scripts** option to create *AdventureWorks2022b*.

Script a database by using the Script option

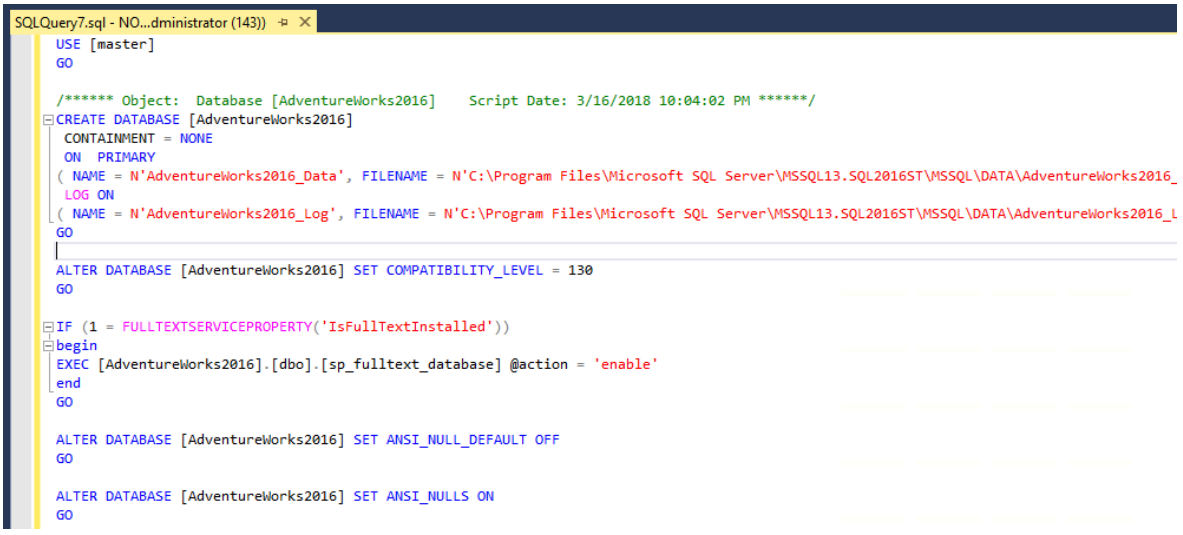
1. Connect to a server that's running SQL Server.

2. Expand the **Databases** node.

3. Right-click the database **AdventureWorks2022** > **Script Database As** > **Create To** > **New Query Editor Window**:



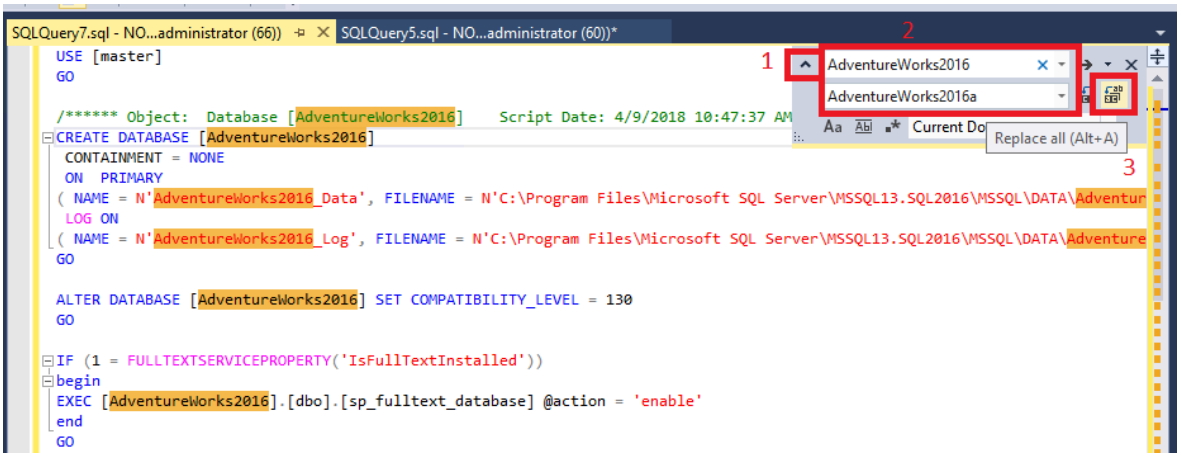
4. Review the database creation query in the window:



This option scripts out only the database configuration options.

5. On your keyboard, select Ctrl+F to open the **Find** dialog box. Select the down arrow to open the **Replace** option. On the top **Find** line, type AdventureWorks2022, and on the bottom **Replace** line, type AdventureWorks2022a.

6. Select **Replace All** to replace all instances of *AdventureWorks2022* with *AdventureWorks2022a*.



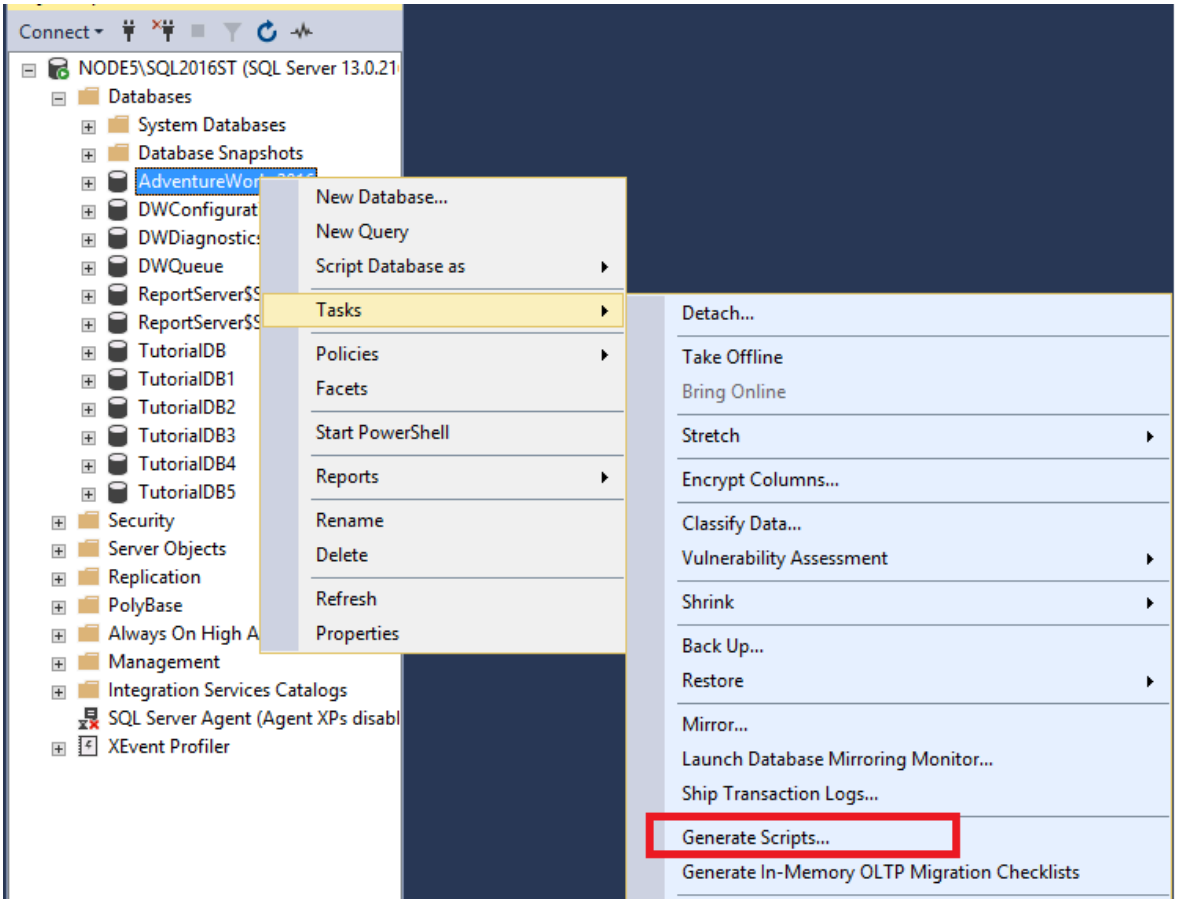
7. Select **Execute** to execute the query and create your new AdventureWorks2022a database.

Script a database by using the Generate Scripts option

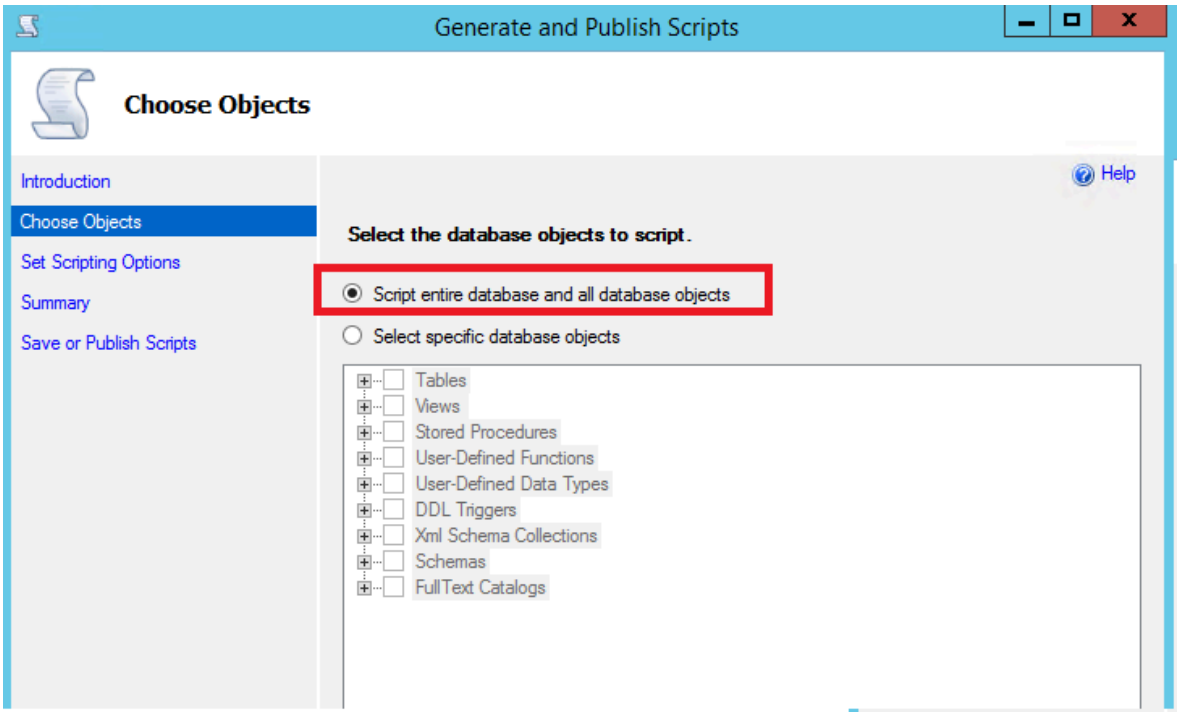
1. Connect to a server that's running SQL Server.

2. Expand the **Databases** node.

3. Right-click **AdventureWorks2022** > **Tasks** > **Generate Scripts**:



4. The **Introduction** page opens. Select **Next** to open the **Chose Objects** page. You can select the entire database or specific objects in the database. Select **Script entire database and all database objects**.

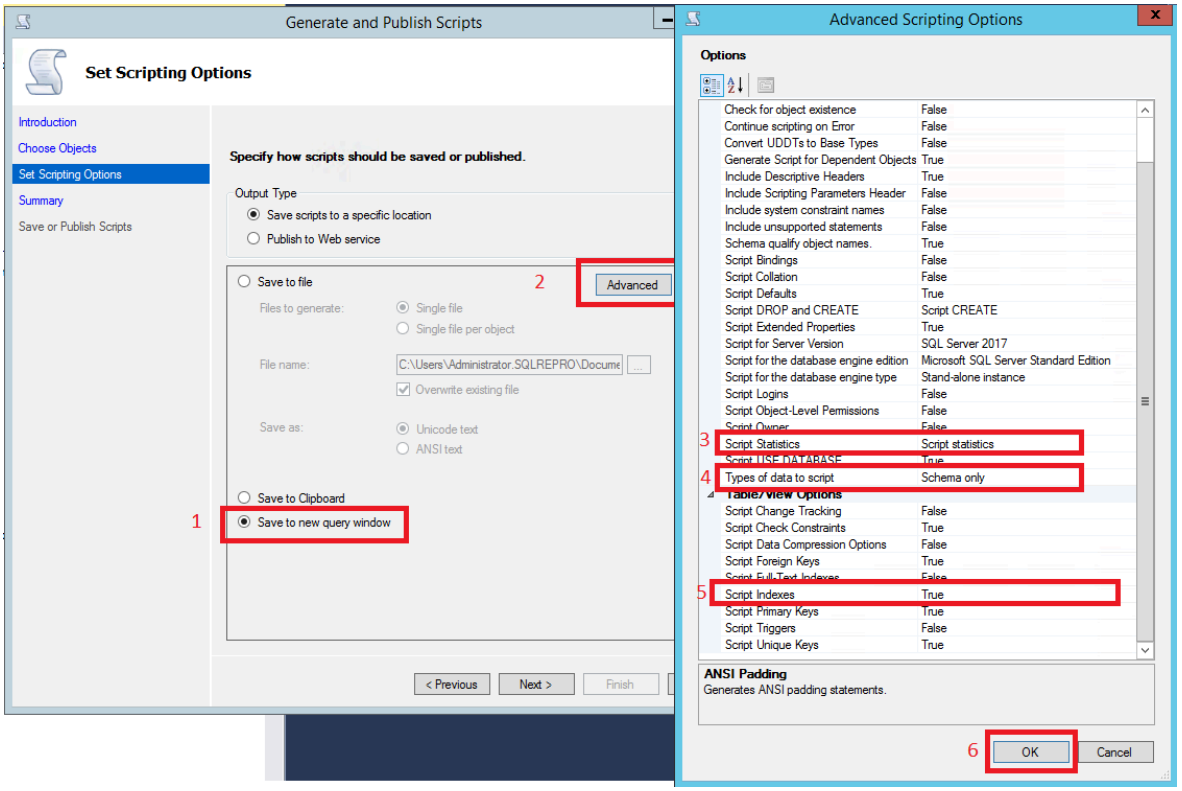


5. Select **Next** to open the **Set Scripting Options** page. Here you can configure where to save the script and some additional advanced options.

   a. Select **Save to new query window**.

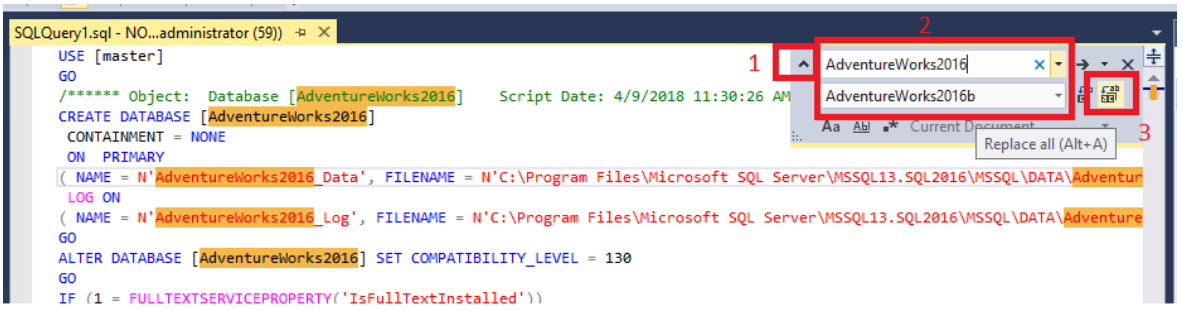
   b. Select **Advanced** and make sure these options are set:

   - **Script Statistics** set to *Script Statistics*.
   - **Types of data to script** set to *Schema only*.
   - **Script Indexes** set to *True*.



> **Note**
>
> You can script the data for the database when you select *Schema and data* for the **Types of data to script** option. However, this isn't ideal with large databases. It can take more memory than SSMS can allocate. This limitation is okay for small databases. If you want to move data for a larger database, use the Import and Export Wizard.

6. Select **OK**, and then select **Next**.
7. Select **Next** on the **Summary**. Then select **Next** again to generate the script in a **New Query** window.
8. On your keyboard, open the **Find** dialog box (Ctrl+F). Select the down arrow to open the **Replace** option. On the top **Find** line, enter *AdventureWorks2022*. On the bottom **Replace** line, enter *AdventureWorks2022b*.
9. Select **Replace All** to replace all instances of *AdventureWorks2022* with *AdventureWorks2022b*.



10. Select **Execute** to execute the query and create your new AdventureWorks2022b database.

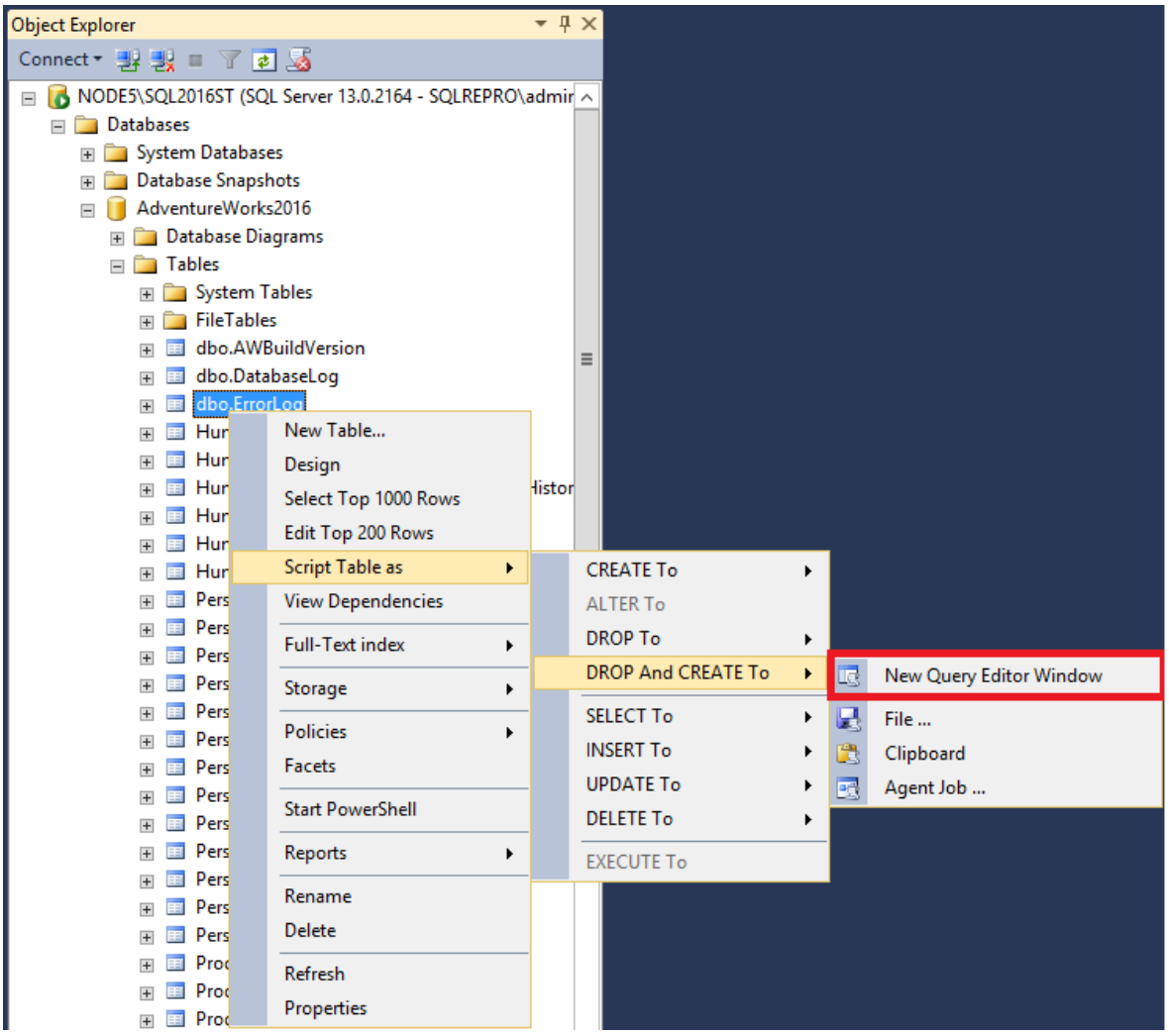
Script tables

This section covers how to script out tables from your database. Use this option to either create the table or drop and create the table. You can also use this option to script the T-SQL associated with modifying the table. An example is to insert into it or update to it. In this section, you drop a table and then re-create it.

1. Connect to a server that's running SQL Server.
2. Expand your **Databases** node.
3. Expand your **AdventureWorks2022** database node.

4. Expand your **Tables** node.

5. Right-click **dbo.ErrorLog** > **Script Table as** > **DROP And CREATE To** > **New Query Editor Window**:



6. Select **Execute** to execute the query. This action drops the *Errorlog* table and re-creates it.

> ⓘ **Note**
>
> The *Errorlog* table is empty by default in the `AdventureWorks2022` database. So you're not losing any data by dropping the table. However, following these steps on a table with data causes data loss.

Script stored procedures

In this section, you learn how to drop and create a stored procedure.

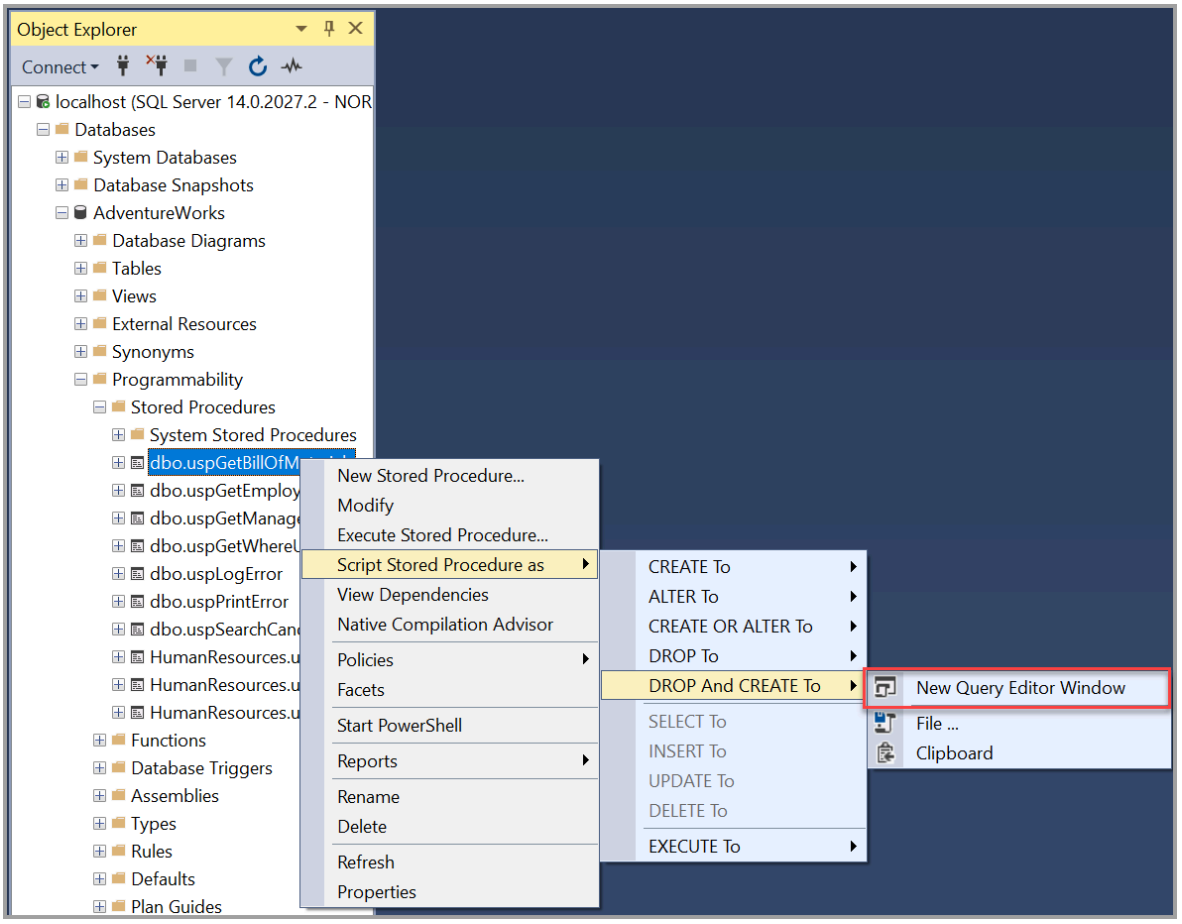
1. Connect to a server that's running SQL Server.

2. Expand your **Databases** node.

3. Expand your **Programmability** node.

4. Expand your **Stored Procedure** node.

5. Right-click the stored procedure **dbo.uspGetBillOfMaterials** > **Script Stored Procedure As** > **DROP and CREATE To** > **New Query Editor Window**:



Script extended events

This section covers how to script out extended events.

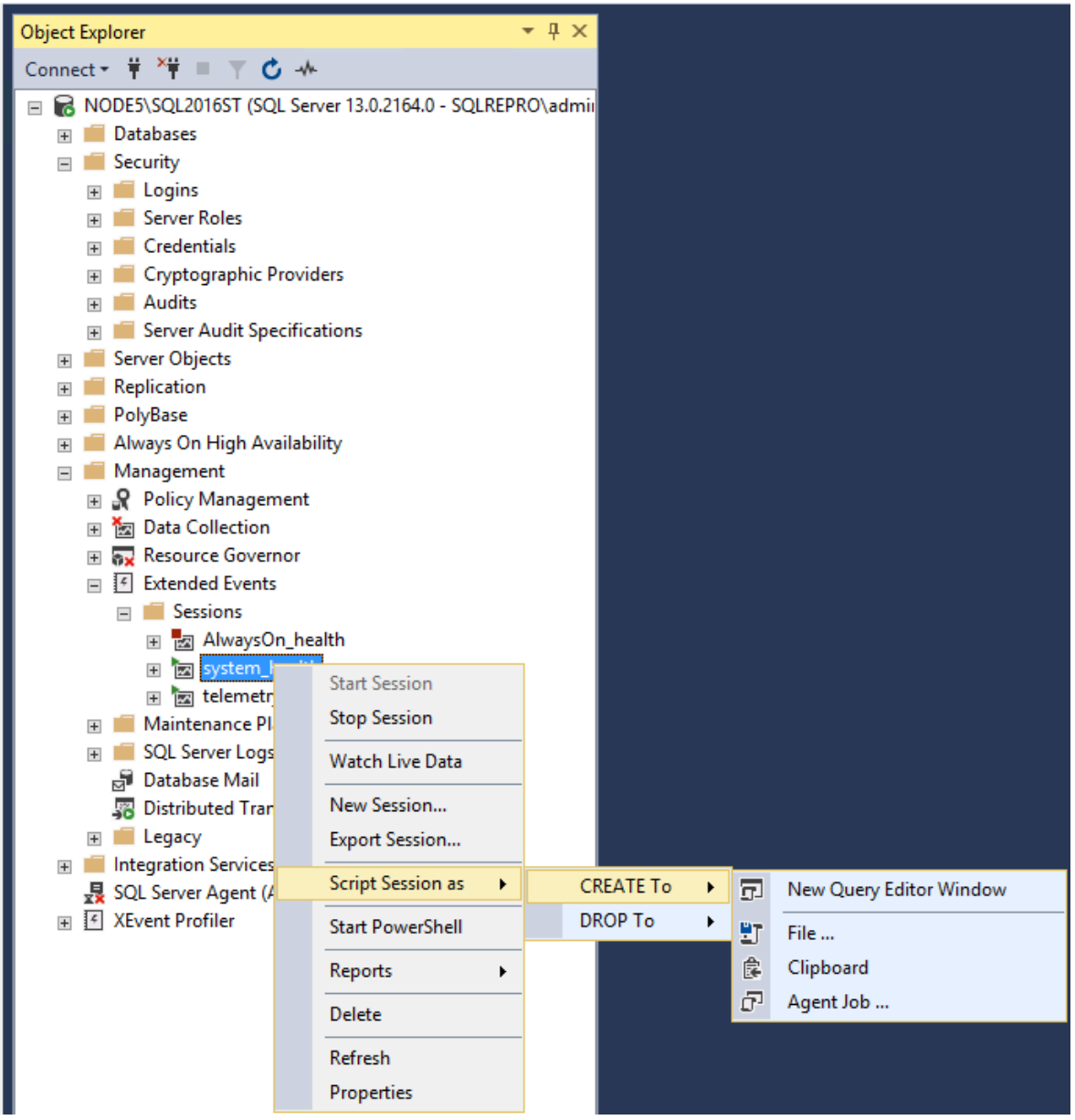
1. Connect to a server that's running SQL Server.

2. Expand your **Management** node.

3. Expand your **Extended Events** node.

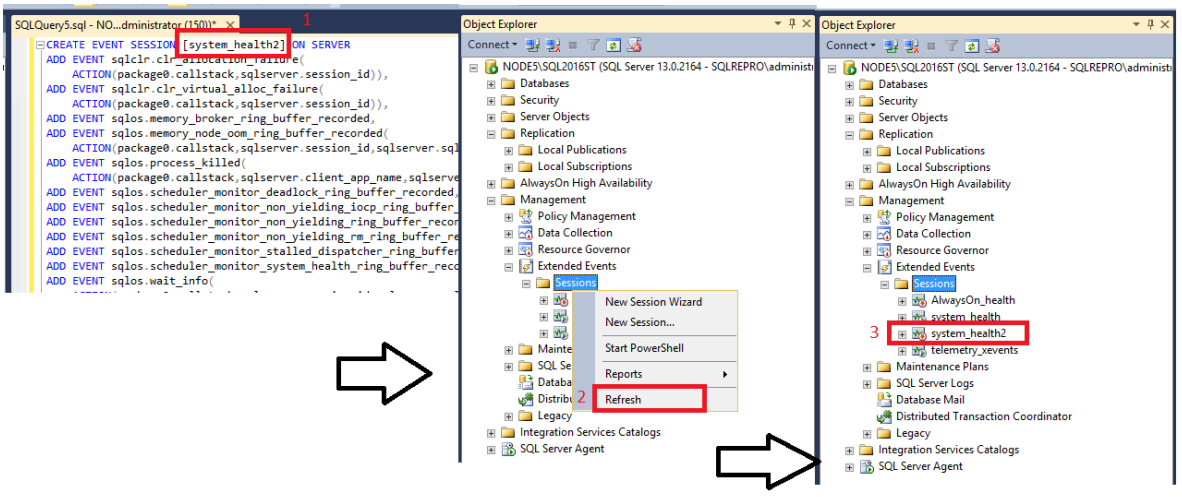
4. Expand your **Sessions** node.

5. Right-click the extended session you're interested in > **Script Session As** > **CREATE To** > **New Query Editor Window**:



6. In the **New Query Editor Window**, modify the new name of the session from *system_health* to *system_health2*. Select **Execute** to execute the query.

7. Right-click **Sessions** in **Object Explorer**. Select **Refresh** to see your new extended event session. The green icon next to the session indicates the session is running. The red icon indicates the session is stopped.



> **Note**
>
> You can start the session by right-clicking it and selecting **Start**. However, this is a copy of the already running **system_health** session, so you can skip this step. You can delete the copy of the extended event session: right-click it and select **Delete**.

Next steps

The best way to get acquainted with SSMS is through hands-on practice. These *tutorial* and *how-to* articles help you with various features available within SSMS. These articles teach you how to manage the components of SSMS and how to find the features that you use regularly.

- Connect to and query an instance
- Using Templates in SSMS
- SSMS Configuration
- Additional Tips and Tricks for using SSMS

Feedback

Was this page helpful? Yes No

Import Flat File to SQL Wizard

Article • 09/18/2024

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance

> For content related to the Import and Export Wizard, see Import and Export Data with the SQL Server Import and Export Wizard.

Import Flat File Wizard is a simple way to copy data from a flat file (.csv, .txt) to a new table in your database. The Import Flat File Wizard supports both comma-separated and fixed width format files. This overview describes the reasons for using this wizard, how to find this wizard, and a simple example to follow.

Why would I use this wizard?

This wizard was created to improve the current import experience leveraging an intelligent framework known as Program Synthesis using Examples (PROSE). For a user without specialized domain knowledge, importing data can often be a complex, error prone, and tedious task. This wizard streamlines the import process as simple as selecting an input file and unique table name, and the PROSE framework handles the rest.

PROSE analyzes data patterns in your input file to infer column names, types, delimiters, and more. This framework learns the structure of the file and does all of the hard work so users don't have to.

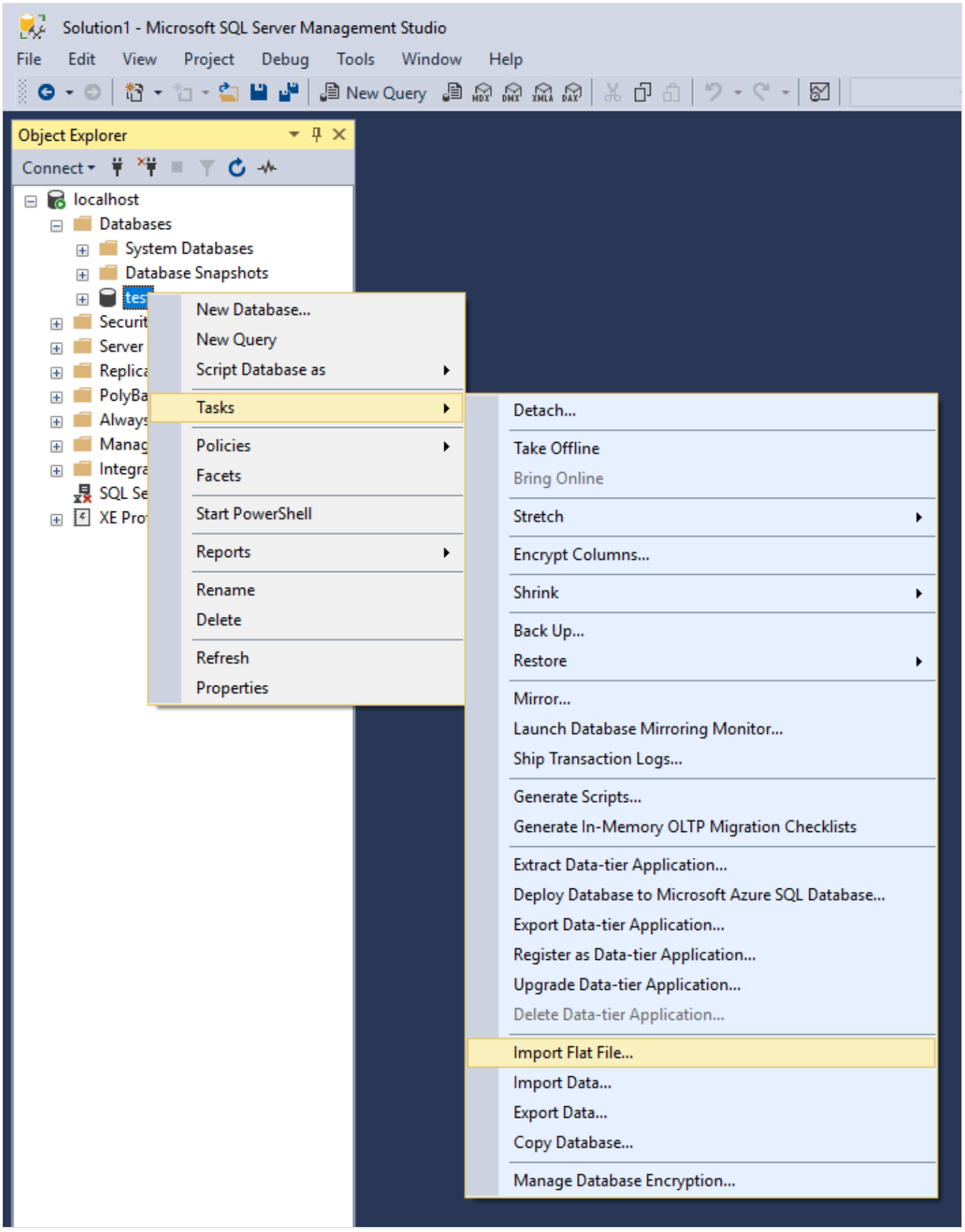
Prerequisites

This feature is available on SQL Server Management Studio (SSMS) v17.3 or later. Make sure you're using the latest version. You can find the latest version here.

Getting Started

To access the Import Flat File Wizard, follow these steps:

1. Open **SQL Server Management Studio**.
2. Connect to an instance of the SQL Server Database Engine or localhost.
3. Expand **Databases**, right-click a database (test in the following example), point to **Tasks**, and select **Import Flat File** above Import Data.

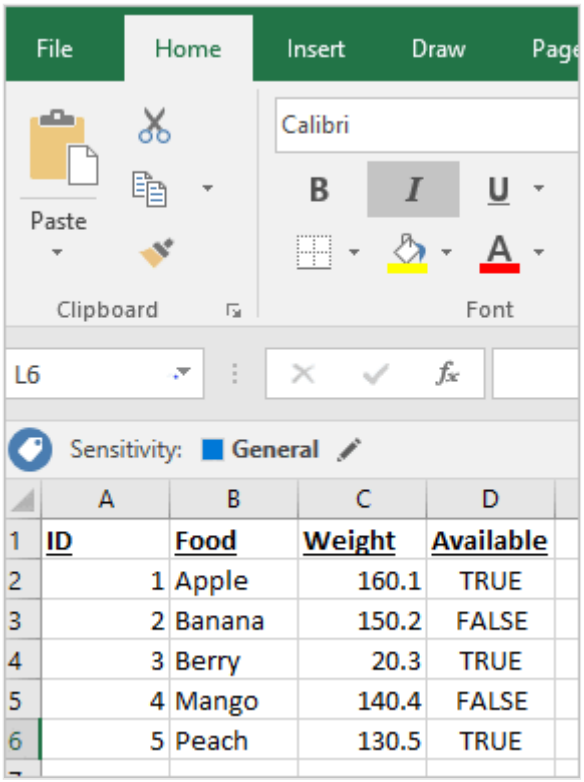


To learn more about the different functions of the wizard, refer to the following tutorial:

Tutorial

For the purposes of this tutorial, feel free to use your own flat file. Otherwise, this tutorial is using the following CSV from Excel, which you're free to copy. If you use this CSV, title it

example.csv and make sure to save it as a csv in an easy location such as your desktop.



	A	B	C	D
1	ID	Food	Weight	Available
2	1	Apple	160.1	TRUE
3	2	Banana	150.2	FALSE
4	3	Berry	20.3	TRUE
5	4	Mango	140.4	FALSE
6	5	Peach	130.5	TRUE

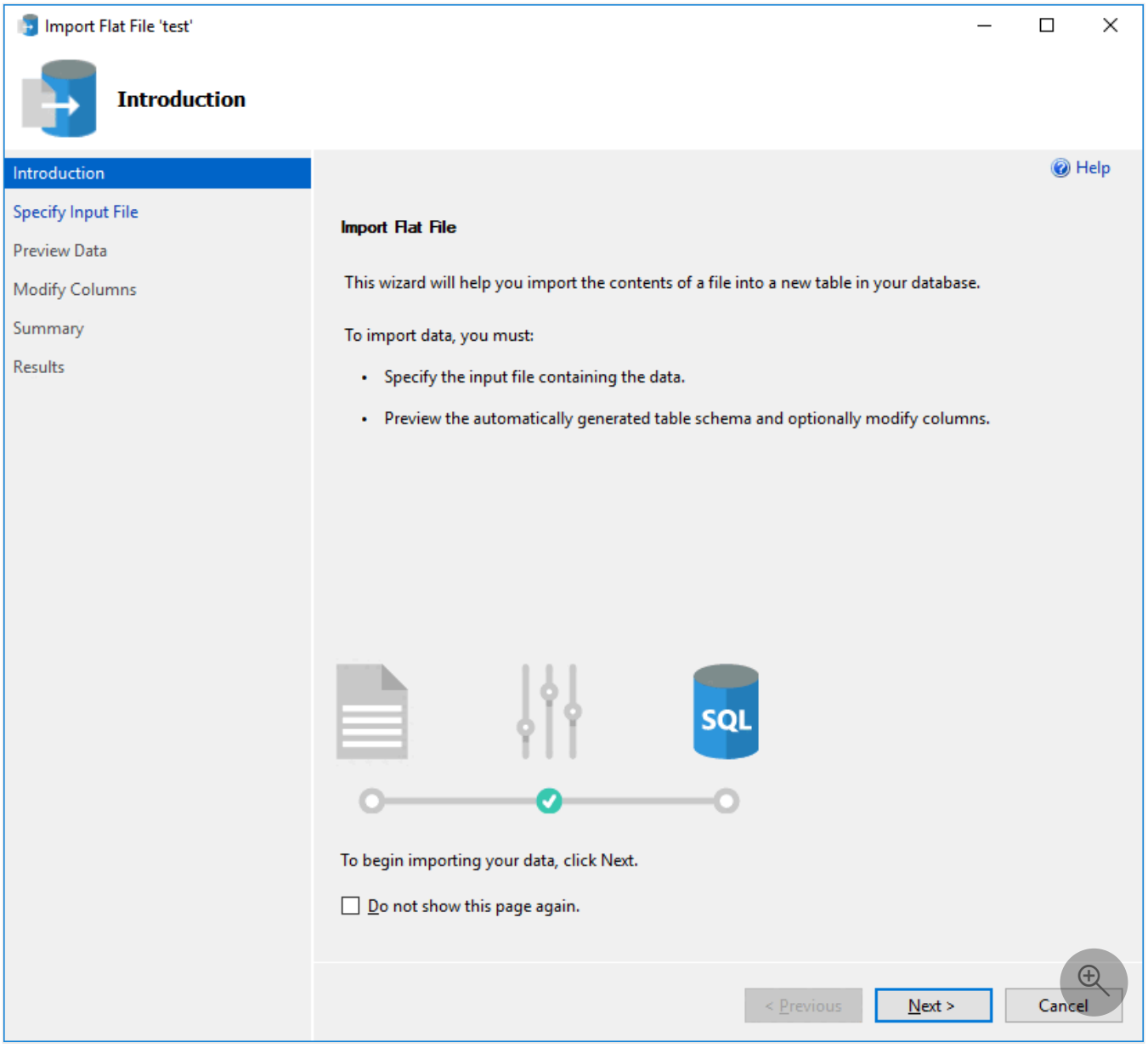
Overview:

1. Access Wizard
2. Specify Input File
3. Preview Data
4. Modify Columns
5. Summary
6. Results

Step 1: Access Wizard and Intro Page

Access the wizard as described here.

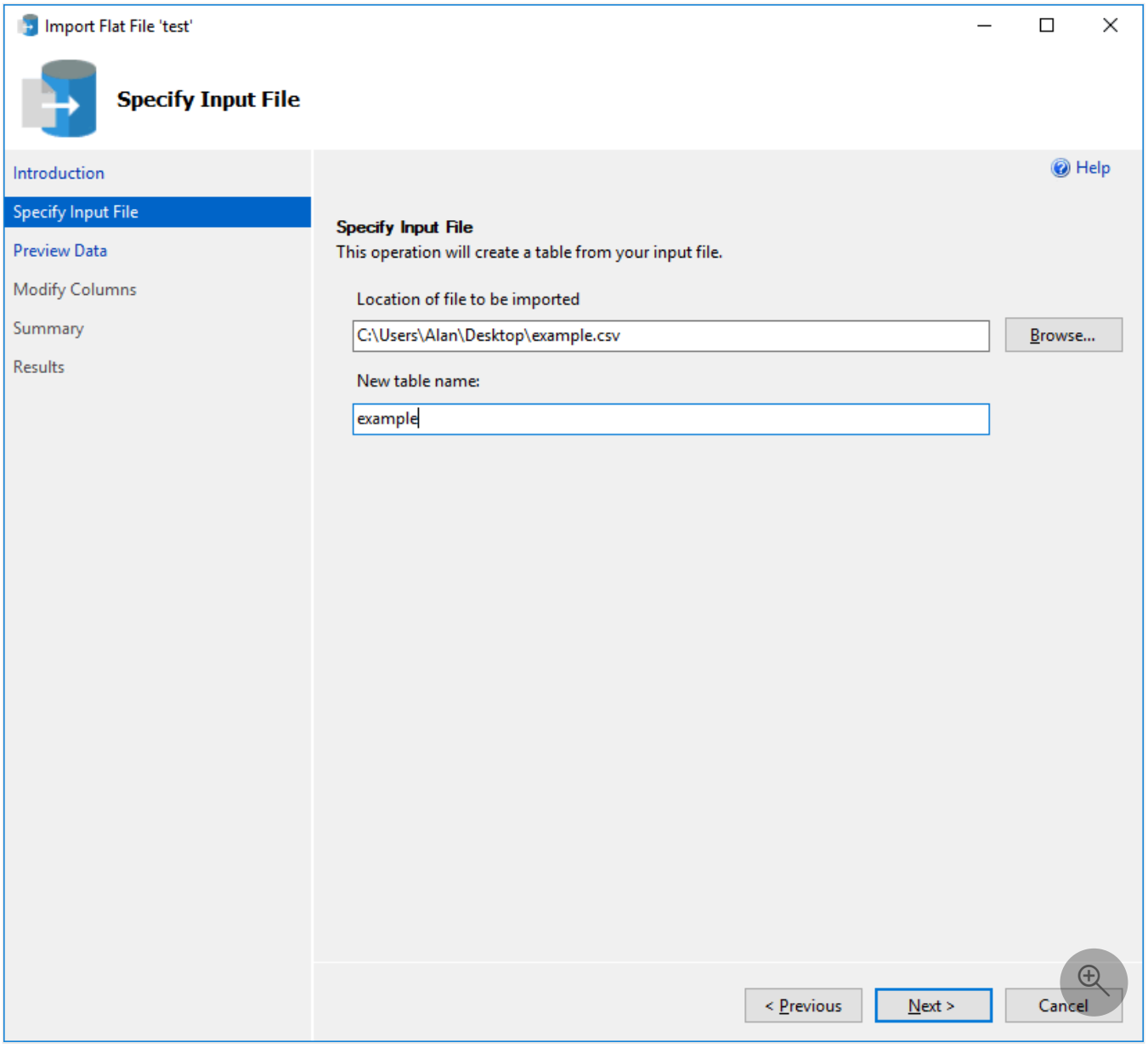
The first page of the wizard is the welcome page. If you don't want to see this page again, feel free to select **Do not show this starting page again.**



Step 2: Specify Input File

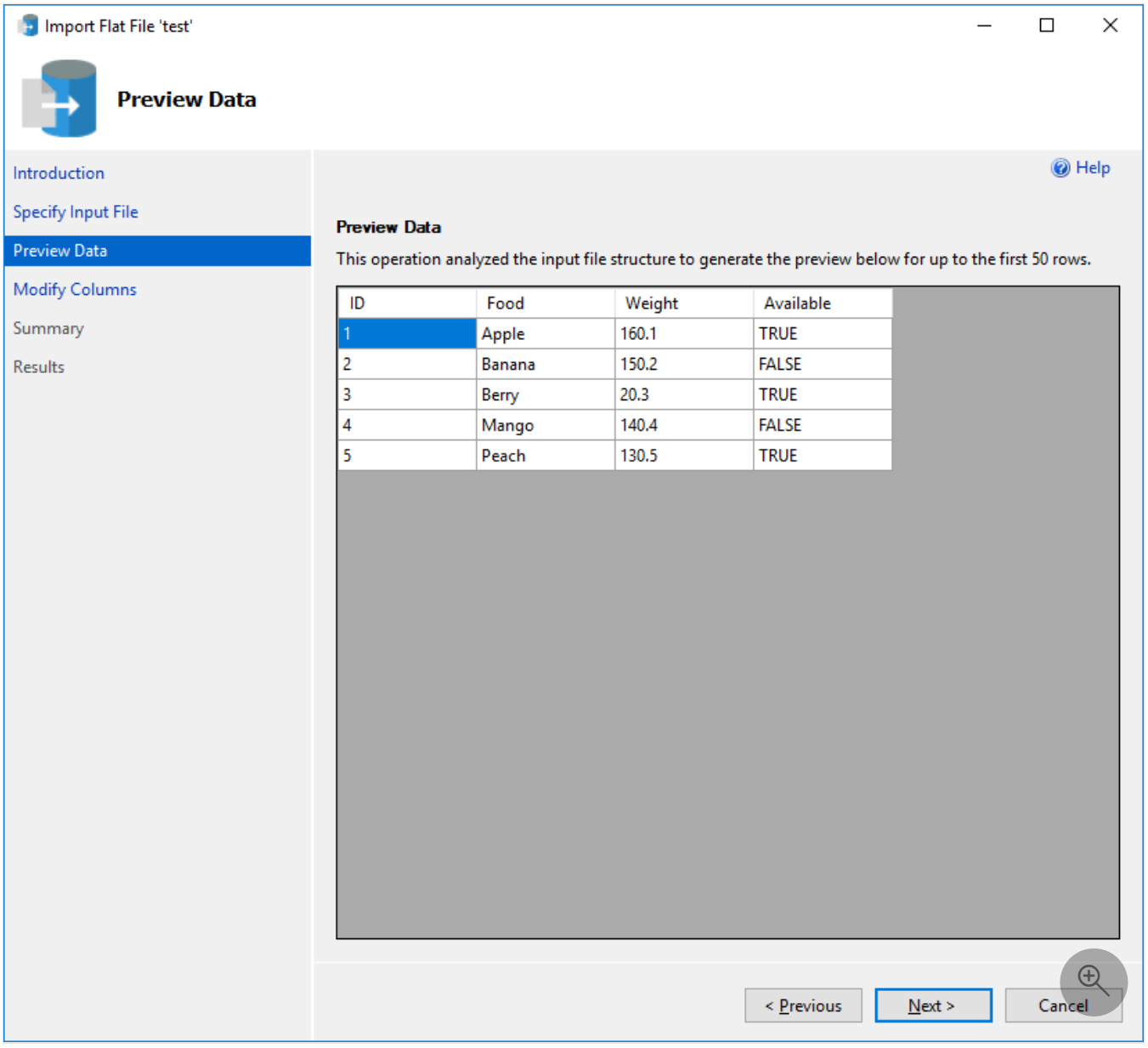
Select browse to select your input file. At default, the wizard searches for .csv and .txt files. PROSE detects if the file is comma-separated or fixed-width format regardless of file extension.

The new table name should be unique, and the wizard doesn't allow you to move further if not.



Step 3: Preview Data

The wizard generates a preview that you can view for the first 50 rows. If there are any problems, select cancel, otherwise proceed to the next page.

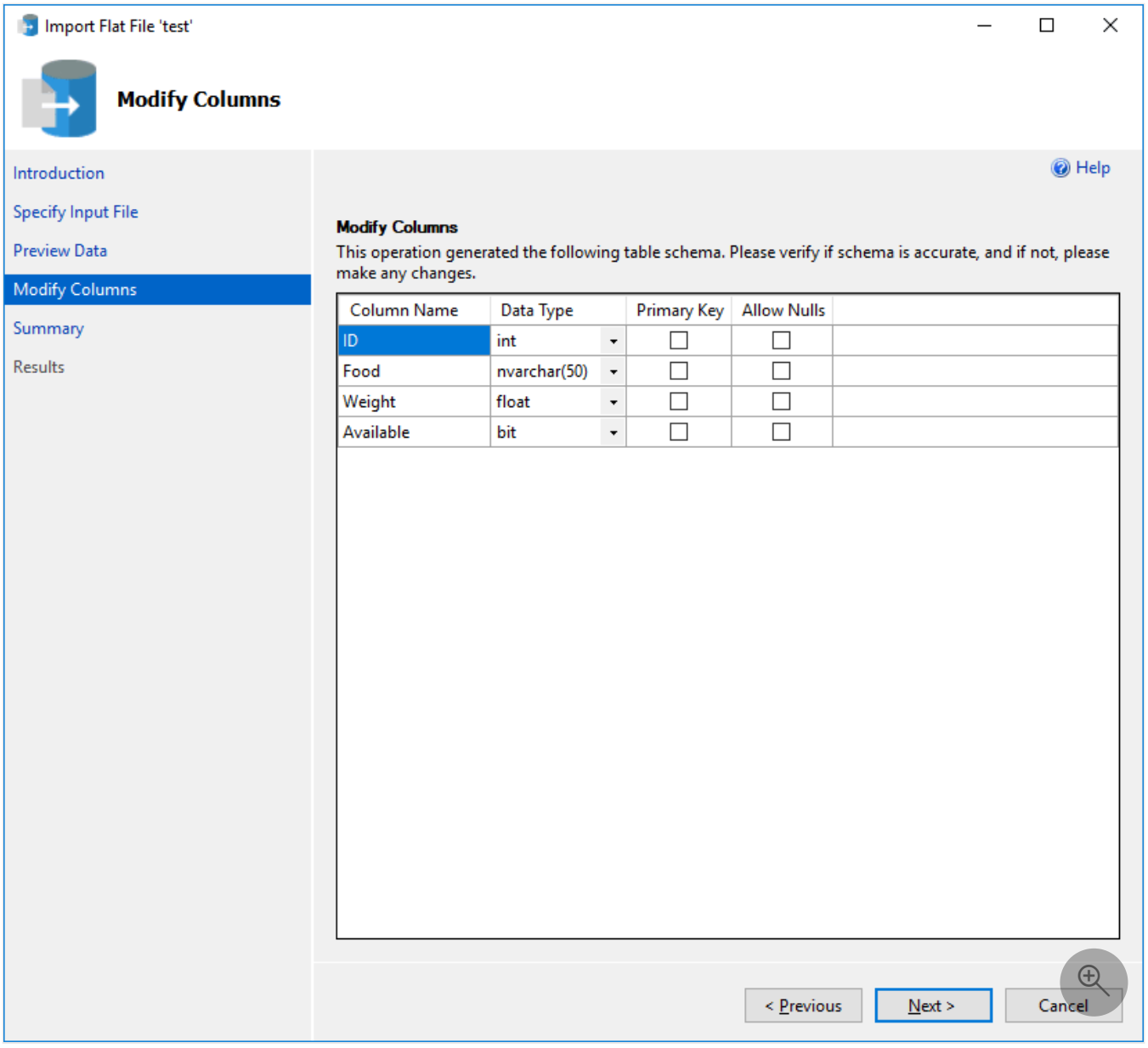


Step 4: Modify Columns

The wizard identifies what it believes are the correct column names, data types, etc. Here's where you can edit the fields if they're incorrect (for example, data type should be a float instead of an int).

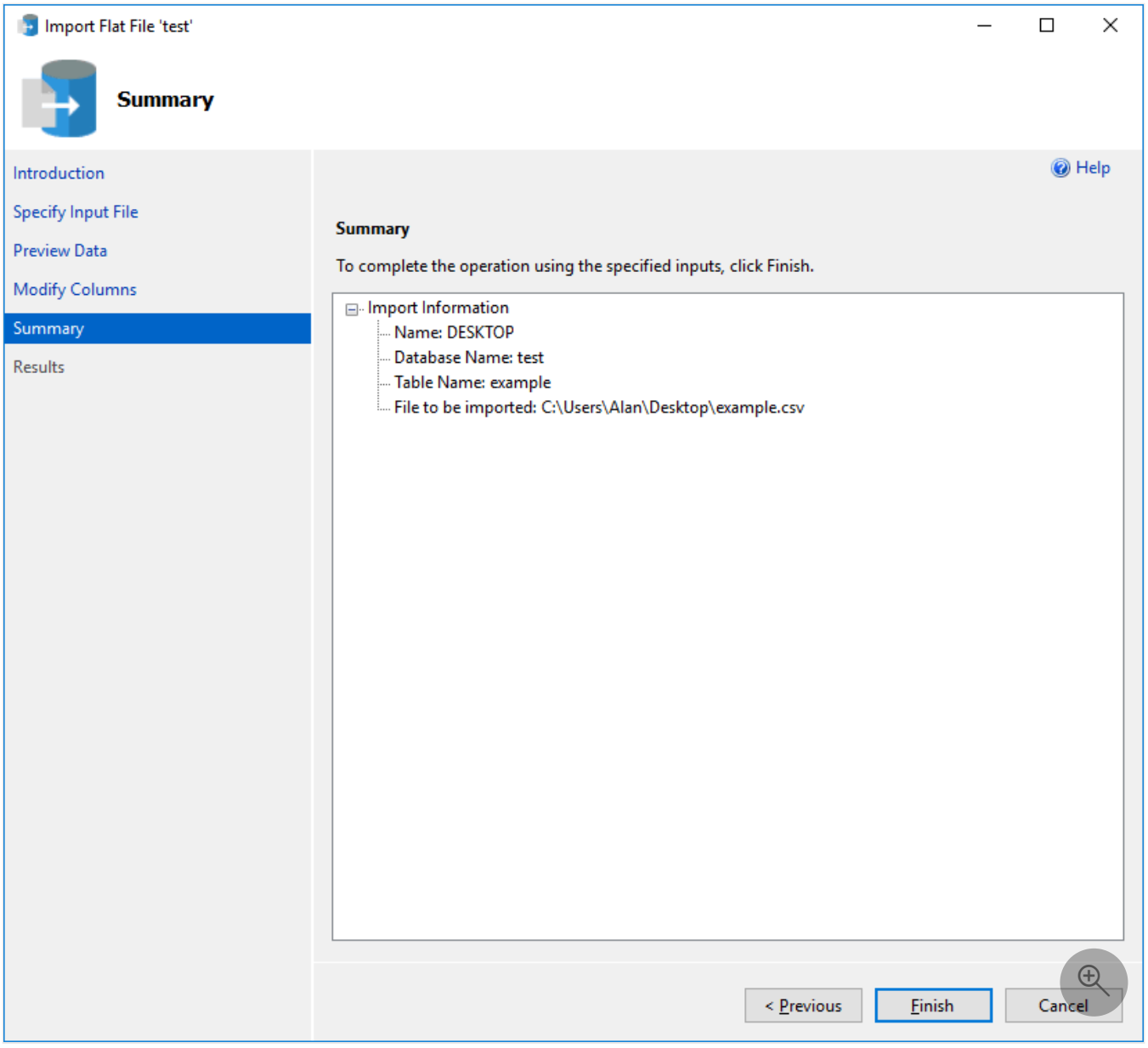
Columns where empty values are detected will have "Allow Nulls" checked. However if you expect nulls in a column and "Allow Nulls" isn't checked, here's where you can update the table definition to allow nulls in one or all columns.

Proceed when ready.



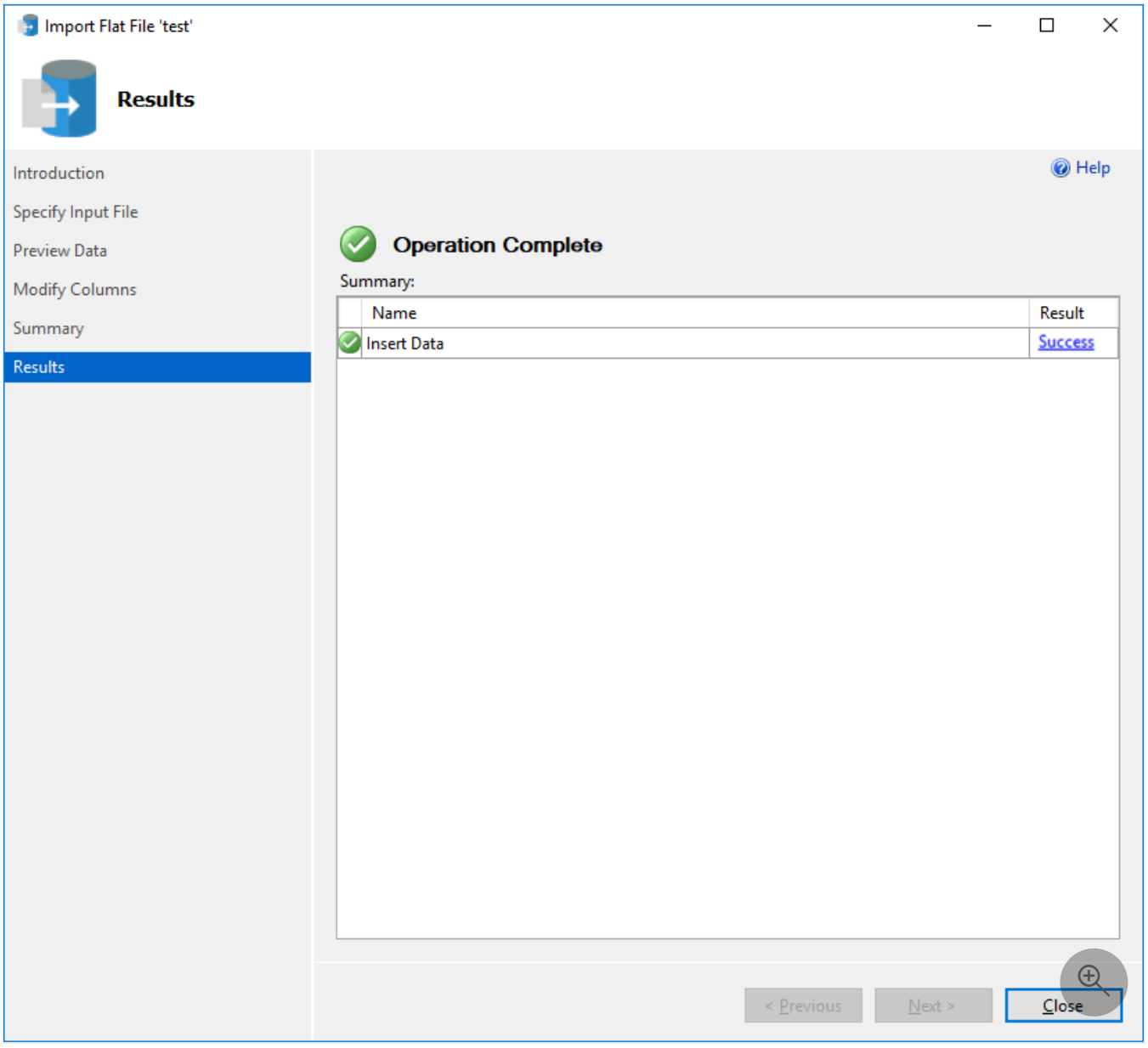
Step 5: Summary

This is simply a summary page displaying your current configuration. If there are issues, you can go back to previous sections. Otherwise, selecting finish attempts the import process.



Step 6: Results

This page indicates whether the import was successful. If a green check mark appears, it was a success, otherwise you might need to review your configuration or input file for any errors.



Troubleshooting

The Import Flat File Wizard detects the data types based on the first 200 rows. In scenarios where data further in the flat file doesn't conform to the automatically detected data types, an error occurs during import. The error message would be similar to the following:

Output

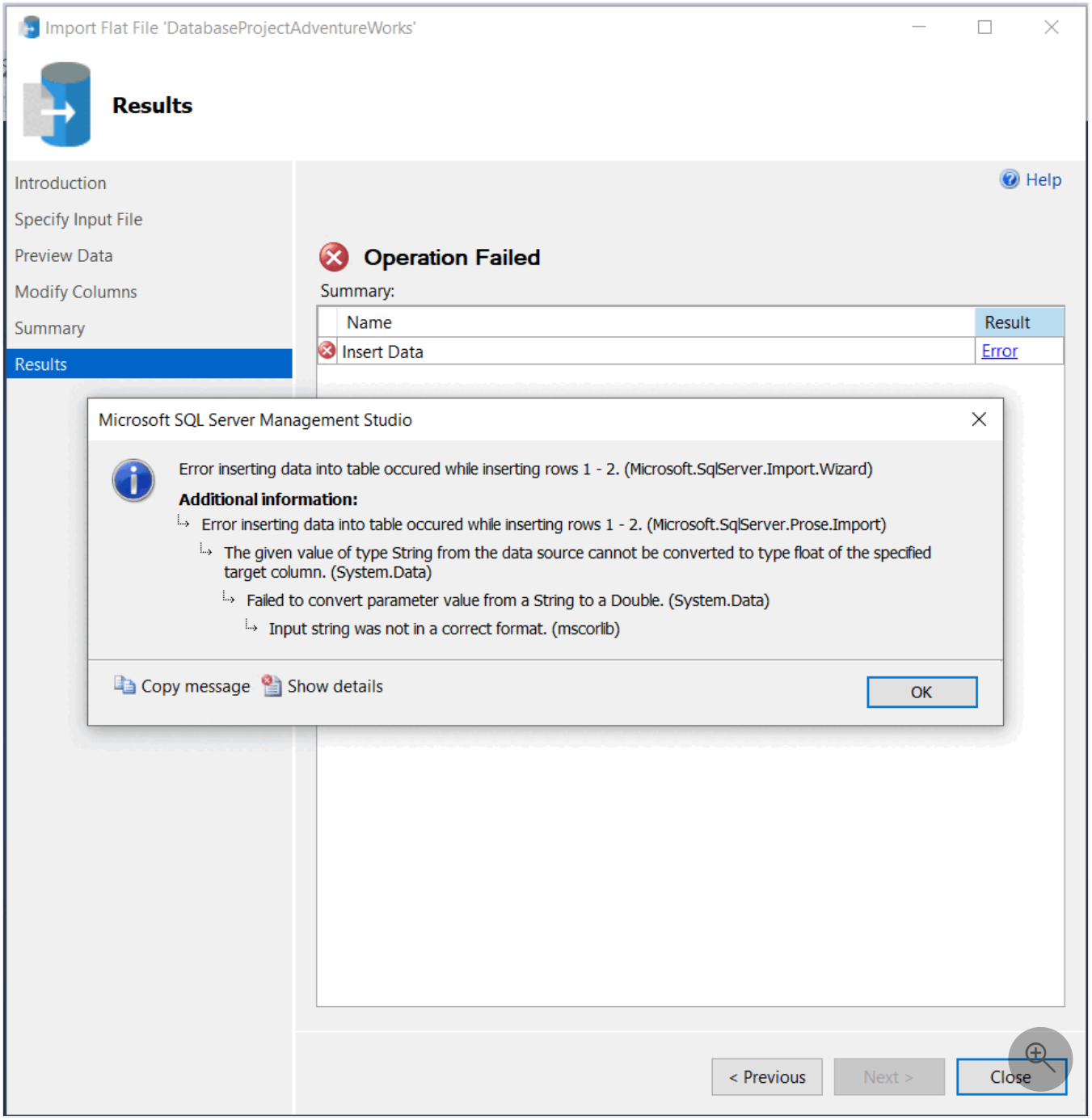
```
Error inserting data into table. (Microsoft.SqlServer.Prose.Import)
The given value of type String from the data source cannot be converted to type
nvarchar of the specified target column. (System.Data)
String or binary data would be truncated. (System.Data)
```

Tactics to alleviate this error:

- Expanding the data type sizes in the Modify Columns step, such as the length of a nvarchar column, might compensate for variations in the data from the remainder of the

flat file.

- Enabling error reporting in the Modify Columns step, especially by a smaller number, will reveal which rows in the flat file contain data that doesn't fit the selected data types. For example, in a flat file where the second row introduces an error, running the import with error reporting with a range of 1 provides a specific error message. Examining the file directly at the location can provide more targeted changes to the data types based on the data in the identified rows.



Output

```
Error inserting data into table occurred while inserting rows 1 - 2.
(Microsoft.SqlServer.Prose.Import)
The given value of type String from the data source cannot be converted to type
```

```
float of the specified target column. (System.Data)
Failed to convert parameter value from a String to a Double. (System.Data)
```

Currently, the importer uses encoding based on the system's active code page. On most machines this defaults to ANSI.

Related content

Learn more about the wizard.

- **Learn more about importing other sources.** If you're looking to import more than flat files, see Import and Export Data with the SQL Server Import and Export Wizard.
- **Learn more about connecting to flat file sources.** If you're looking for more information about connecting to flat file sources, see Connect to a Flat File Data Source (SQL Server Import and Export Wizard).
- **Learn more about PROSE.** If you're looking for an overview of the intelligent framework used by this wizard, see PROSE SDK.

Connection pooling in SQL Server Management Studio

Article • 03/11/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Opening connections to a SQL Server can be an expensive operation, therefore SQL Server Management Studio utilizes the Connection Pooling feature of the ADO.NET SqlClient driver. For more information, see SQL Server connection pooling (ADO.NET).

This article provides more information on connection pooling in SQL Server Management Studio, and ways to mitigate the effect it can have on your server.

Usage and benefits

SQL Server Management Studio is a complex application with numerous features, many of which require information from a database or server. Much of this information is loaded on-demand to avoid overhead when making an initial connection, and to avoid unnecessary work if a feature isn't being used.

Connection pooling can help reduce the overhead of retrieving this information. Features in SQL Server Management Studio generally use the same base connection entered by the user in the connection dialog, and different features can reuse the same physical connection instead of opening a new one.

Non-pooled connections

Not all connections in SQL Server Management Studio are pooled. Some, such as the connection used for each query editor, are explicitly not pooled. There are several reasons for this, including the need to keep a specific session ID (SPID) associated with the feature, or to ensure that changes made to the connection properties stay consistent across usage.

Control open connections

While connection pooling improves performance, it also results in connections staying open for longer than might seem necessary. When a connection is returned to the pool,

it stays open, but in an idle (or sleeping) state. This state can prevent actions from being taken that require all connections to be closed, such as dropping or altering a database.

There are a few options available to close these idle connections:

- Wait until ADO.NET closes the connections. This happens for connections which haven't been used for between four and eight minutes.
- Some operations in SQL Server Management Studio (such as `DROP DATABASE`) provide the option to close all existing connections before performing the operation.
- Use the **Close Idle SQL Connections** command under the **Help** menu. This option immediately closes all idle connections for the current instance of SQL Server Management Studio. Active connections aren't affected except for immediately being closed when they're done being used instead of being returned to the pool.

  > **Note**
  >
  > Closing idle connections can result in decreased performance the next time a new connection to the server is needed, because the connection needs to be reestablished.

- Close SQL Server Management Studio. Closing SQL Server Management Studio immediately closes all open connections associated with that database or instance.
- Run the KILL statement to close any sessions that are blocking your operation.

Related content

- SQL Server connection pooling (ADO.NET)

Feedback

Was this page helpful? Yes No

Tool windows in SQL Server Management Studio

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

SQL Server Management Studio provides many powerful tool windows for all phases of development and administration. Some tools can be used on any SQL Server component, and others are for certain components only. The following table identifies the tools that can be used for all components of SQL Server.

Expand table

Tool	Purpose
Object Explorer	Browse servers, create and locate objects, manage data sources, and view logs. This tool is accessed from the **View** menu.
Solution Explorer	Store and organize scripts and related connection information in projects called SQL Server Scripts. You can store several SQL Server Scripts as Solutions and use source control to manage scripts as they evolve over time. This tool is accessed from the **View** menu.
Template Explorer	Create queries based on existing templates. You can also create your custom queries or alter the existing templates to fit your scenarios. This tool is accessed from the **View** menu.
Dynamic Help	Show a list of related Help topics as you click on a component or type code.

The tools in SQL Server Management Studio work together.

For example, you can:

- Register a server with Object Explorer.
- Open a SQL Editor window connected to a specific database from Object Explorer.

See Also

Use SQL Server Management Studio

Feedback

Was this page helpful? Yes No

Build Database Projects by Using SQL Server Management Studio

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

A database script project is an organized set of scripts, connection information, and templates that are all associated with a database or one part of a database. Microsoft SQL Server provides the SQL Server Management Studio for administering and designing SQL Server databases within the context of a script project. SQL Server Management Studio includes designers, editors, guides and wizards to assist users in developing, deploying and maintaining databases.

SQL Server Management Studio

SQL Server Management Studio is a suite of administrative tools for managing the components belonging to SQL Server. This integrated environment allows users to perform a variety of tasks, such as backing up data, editing queries, and automating common functions within a single interface.

SQL Server Management Studio includes the following tools:

- Code Editor is a rich script editor for writing and editing scripts. SQL Server Management Studio provides four versions of the Code Editor; the Database Engine Query Editor for Transact-SQL scripts, the DMX Query Editor, the MDX Query Editor, and the XML/A Query Editor.
- Object Explorer for locating, modifying, scripting or running objects belonging to instances of SQL Server.
- Template Explorer for locating and scripting templates.
- Solution Explorer for organizing and storing related scripts as parts of a project.
- Properties Window for displaying the current properties of selected objects.

SQL Server Management Studio supports efficient work processes by providing:

- Disconnected access. You can write and edit scripts without connecting to an instance of SQL Server.

- Scripting from any dialog box. You can create a script from any dialog box so that you can read, modify, store and reuse the scripts after you create them.
- Nonmodal dialog boxes. When you access a UI dialog box you can browse other resources in SQL Server Management Studio without closing the dialog box.

Solutions and Script Projects

Solution Explorer is a utility to store and reopen database solutions. Solutions organize related script projects and files. Script projects store SQL Server script files, SQL templates, connection information and other miscellaneous files. When a script is saved in a script project, users are able to:

- Maintain version control on scripts.
- Store results options with a script.
- Organize related scripts in a single script project.
- Save connection information with scripts.

Solution Explorer is a tool for developers who are creating and reusing scripts that are related to the same project. If a similar task is required later, you can use group of scripts that were stored in a project. If you have created applications by using Microsoft Visual Studio, you will find Solution Explorer very familiar.

A solution consists of one or more script projects. A project consists of one or more scripts or connections. A project may also include nonscript files.

See Also

Use SQL Server Management Studio
Solutions (SQL Server Management Studio)

Feedback

Was this page helpful? Yes No

Understand SQL Server Management Studio Windows management

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The tool windows in SQL Server Management Studio are part of a highly functional, flexible, and efficient system that allows you to:

- Maximize the user workspace for development and management.
- Reduce the number of unused windows displayed at one time.
- Easily customize the user environment.

Manipulating windows is central to the Management Studio environment. Users can easily access the tools and windows they use frequently. Users can control how much space they want to allocate to different information, and the environment should maximize the space available for editing queries accordingly. Windows can be moved to different locations on the screen. Many windows can be undocked and dragged out of the Management Studio frame, which is useful when using more than one monitor.

You can increase your editing space while maintaining functionality with the Auto Hide feature. This feature displays the window as a tab within a bar along the border of the main Management Studio environment. When the pointer is placed over one of these tabs, the underlying window reveals itself. Auto Hide for a window can be toggled by selecting the **Auto Hide** button, represented by a pushpin in the upper-right corner of the window. There's also an **Auto Hide All** option on the **Window** menu.

Some components can be configured in either tabbed mode where components appear as tabs in the same docking location, or in multiple document interface (MDI) mode where each document has its own window. To configure this feature, navigate to **Tools** > **Options** > **Environment**, and select **General**.

Cached identity information

When a login (or a contained database user) connects and is authenticated, the connection caches identity information about the login. For Windows Authentication, information about membership in Windows groups is included. The identity remains authenticated as long as the connection is maintained. To force changes in the identity, such as a password reset or change in Windows group membership, the account must

sign out from the authentication authority (Windows or SQL Server), and sign in again. A member of the **sysadmin** fixed server role, or any login with the `ALTER ANY CONNECTION` permission, can use the `KILL` command to end a connection and force a login to reconnect. SQL Server Management Studio can reuse connection information when opening multiple connections to Object Explorer and Query Editor windows. To force reconnection, close all connections.

Related content

- Use SQL Server Management Studio
- The SQL Server Management Studio Environment

Feedback

Was this page helpful? Yes No

Administer servers with SQL Server Management Studio

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

SQL Server Management Studio (SSMS) is a rich, integrated administrative client designed to meet the SQL Server and Azure SQL Database administrator's server management requirements. In SSMS, administrative tasks are accomplished using Object Explorer, which allows you to connect to any server in the SQL Server family and graphically browse its contents. A server can be an instance of the Database Engine, Analysis Services (SSAS), Reporting Services (SSRS), Integration Services (SSIS), Azure SQL Managed Instance, or Azure SQL Database.

The tool components of SSMS include Registered Servers, Object Explorer, Solution Explorer, Template Explorer, the Object Explorer Details page, and the document window. To display a tool, on the **View** menu, select the tool name. To display the Query Editor tool, select the **New Query** button on the toolbar.

To get started, download SQL Server Management Studio.

Encrypt your connection

In SQL Server 2019 (15.x) and earlier versions, network traffic between SSMS and SQL Server is unencrypted by default. Don't work with sensitive data (including passwords) in SSMS unless you establish an encrypted connection. For more information, see How to: Enable Encrypted Connections to the Database Engine (SQL Server Configuration Manager).

For more information about encryption in SQL Server Management Studio 20 and later versions, see Release notes for SQL Server Management Studio (SSMS).

SSMS features

Use SSMS to:

- Register servers
- Connect to an instance of the Database Engine, SSAS, SSRS, SSIS, Azure SQL Managed Instance, or Azure SQL Database

- Configure server properties
- Manage database and SSAS objects such as cubes, dimensions, and assemblies
- Create objects, such as databases, tables, cubes, database users, and logins
- Manage files and filegroups
- Attach or detach databases
- Launch scripting tools
- Manage security
- View system logs
- Monitor current activity
- Configure replication
- Manage full-text indexes

To start and stop SQL Server or SQL Server Agent, use SQL Server Configuration Manager.

Related content

- Download SQL Server Management Studio (SSMS)
- Use SQL Server Management Studio
- How to: View server properties (SQL Server Management Studio)

Feedback

Was this page helpful? Yes No

Arguments for External Tools

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Arguments are variables that the Studio environment supplies values for when an external tool is launched from the **Tools** menu. External tools such as Notepad can be added to the **Tools** menu using the **External Tools** dialog box.

The following table lists arguments for external tools.

Expand table

Name	Argument	Description
Item Path	$(ItemPath)	The complete file name of the current source (defined as drive + path + file name); blank if a non-source window is active.
Item Directory	$(ItemDir)	The directory of the current source (defined as drive + path); blank if a non-source window is active.
Item File Name	$(ItemFilename)	The file name of the current source (defined as file name); blank if a non-source window is active.
Item extension	$(ItemExt)	The file name extension of the current source.
Current Line*	$(CurLine)	The current line position of the cursor in the editor.
Current Column*	$(CurCol)	The current column position of the cursor in the editor.
Current Text*	$(CurText)	The current text (the word under the current cursor position, or a single-line selection, if there is one).
Target Path	$(TargetPath)	The complete file name of the target (defined as drive + path + file name).
Target Directory	$(TargetDir)	The directory of the target.
Target Name	$(TargetName)	The file name of the target.
Target Extension	$(TargetExt)	The file name extension of the target.

Name	Argument	Description
Project Directory	$(ProjDir)	The directory of the current project (defined as drive + path).
Project File Name	$(ProjFileName)	The file name of the current project (defined as drive + path + file name).
Solution Directory	$(SolutionDir)	The directory of the current solution (defined as drive + path).
Solution File Name	$(SolutionFileName)	The file name of the current solution (defined as drive + path + file name).

*The current line, current column, or current text is based on the position of the cursor in the text editor as shown in the status bar.

See Also

External Tools Dialog Box
General User Interface Elements

Feedback

Was this page helpful? Yes No

Add an External Tool to the Tools Menu (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can launch any Microsoft Windows or Microsoft .NET Framework application from SQL Server Management Studio. External applications can be added to, and run from, the **Tools** menu.

To add an external tool to the Tools menu

1. On the **Tools** menu, click **External Tools**.
2. In the **Title** text box, type the name you want to appear in the **Menu contents** list.
3. In the **Command** text box, type the program name. Include the path to the executable file if necessary.
4. In the **Arguments** text box, type the program arguments if necessary.
5. In the **Initial directory** text box, type the program's initial directory if necessary.
6. To add the tool to the **Menu contents** list, click **Add**; and then click **OK**.

Feedback

Was this page helpful? Yes No

SSMS Utility

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The **SSMS** utility opens SQL Server Management Studio. If specified, **Ssms** also establishes a connection to a server, and opens queries, scripts, files, projects, and solutions.

You can specify files that contain queries, projects, or solutions. Files that contain queries are automatically connected to a server if connection information is provided and the file type is associated with that type of server. For instance, .sql files open a SQL Query Editor window in SQL Server Management Studio, and .mdx files open an MDX Query Editor window in SQL Server Management Studio. **SQL Server Solutions and Projects** open in SQL Server Management Studio.

> **Note**
>
> The **Ssms** utility does not run queries. To run queries from the command line, use the **sqlcmd** utility.

Syntax

```
syntaxsql
```

```
Ssms
[scriptfile] [projectfile] [solutionfile]
[-S servername] [-d databasename] [-G] [-U username] [-E] [-nosplash] [-log
[filename]?] [-?]
```

Arguments

scriptfile Specifies one or more script files to open. The parameter must contain the full path to the files.

projectfile Specifies a script project to open. The parameter must contain the full path to the script project file.

solutionfile Specifies a solution to open. The parameter must contain the full path to the solution file.

[-**S** *servername*] Server name

[-**d** *databasename*] Database name

[-**G**] Connect using Active Directory Authentication. The type of connection is determined whether -**U** is included.

> ⓘ Note
>
> **Active Directory - Universal with MFA support** is not currently supported.

[-**U** *username*] User name when connecting with 'SQL Authentication'

> ⓘ Note
>
> -**P** was removed in SSMS version 18.0.
>
> Workaround: Try to connect to the server once using the UI and save your password.

[-**E**] Connect using Windows Authentication

[-**nosplash**] Prevents SQL Server Management Studio from displaying the splash screen graphic while opening. Use this option when connecting to the computer running SQL Server Management Studio by means of Terminal Services over a connection with a limited bandwidth. This argument is not case-sensitive and may appear before or after other arguments

[-**log***[filename]?*] Logs SQL Server Management Studio activity to the specified file for troubleshooting

[-**?**] Displays command-line help

Remarks

All of the switches are optional and separated by a space except files, which are separated by commas. If you do not specify any switches, **Ssms** opens SQL Server Management Studio as specified in the **Options** settings on the **Tools** menu. For

example, if the **Environment/General** page **At startup** option specifies **Open new query window**, **Ssms** opens with a blank Query Editor.

The **-log** switch must appear at the end of the command line, after all, other switches. The filename argument is optional. If a filename is specified, and the file does not exist, the file is created. If the file cannot be created - for example, due to insufficient write access, the log is written to the nonlocalized APPDATA location instead (See below). If the filename argument is not specified, two files are written to the current user's nonlocalized application data folder. The nonlocalized application data folder for SQL Server can be found from the APPDATA environment variable. For example, for SQL Server 2012, the folder is <system drive>:\Users\<username>\AppData\Roaming\Microsoft\AppEnv\10.0\. The two files are, by default, named ActivityLog.xml and ActivityLog.xsl. The former contains the activity log data, and the latter is an XML style sheet, which provides a more convenient way to view the XML file. Use the following steps to view the log file in your default XML viewer, like Internet Explorer: Click Start, then click Run...", then type "<system drive>:\Users\<username>\AppData\Roaming\Microsoft\AppEnv\10.0\ActivityLog.xml" into the field provided, and then press Enter.

Files that contain queries prompt to be connected to a server if connection information is provided and the file type is associated with that type of server. For instance, .sql files open a SQL Query Editor window in SQL Server Management Studio, and .mdx files open an MDX Query Editor window in SQL Server Management Studio. **SQL Server Solutions and Projects** open in SQL Server Management Studio.

The following table maps server types to file extensions.

Expand table

Server type	Extension
SQL Server	.sql
SQL Server Analysis Services	.mdx .xmla

Examples

The following script opens SQL Server Management Studio from a command prompt with the default settings:

```
Console
```

```
Ssms
```

The following scripts opens SQL Server Management Studio from a command prompt using *Active Directory - Integrated*:

```
Console
Ssms.exe -S servername.database.windows.net -G
```

The following script opens SQL Server Management Studio from a command prompt, with Windows Authentication, with the Code Editor set to the server `ACCTG` and the database `AdventureWorks2022`, without showing the splash screen:

```
Console
Ssms -E -S ACCTG -d AdventureWorks2022 -nosplash
```

The following script opens SQL Server Management Studio from a command prompt, and opens the MonthEndQuery script.

```
Console
Ssms "C:\Documents and Settings\username\My Documents\SQL Server Management
Studio Projects\FinanceScripts\FinanceScripts\MonthEndQuery.sql"
```

The following script opens SQL Server Management Studio from a command prompt, and opens the NewReportsProject project on the computer named `developer`:

```
Console
Ssms "\\developer\fin\ReportProj\ReportProj\NewReportProj.ssmssqlproj"
```

The following script opens SQL Server Management Studio from a command prompt, and opens the MonthlyReports solution:

```
Console
Ssms "C:\solutionsfolder\ReportProj\MonthlyReports.ssmssln"
```

See Also

Feedback

Was this page helpful? Yes No

SQL Server Management Studio keyboard shortcuts

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

SQL Server Management Studio (SSMS) offers keyboard shortcuts. By default, it uses the SQL Server scheme, with keyboard shortcuts based on Visual Studio. To change the keyboard scheme or add more keyboard shortcuts, on the **Tools** menu, select **Options**. Select the desired keyboard scheme on the **Environment**, **Keyboard** page.

> **Note**
>
> To show only the headings, select **Collapse All** at the top of this page.

Menu Activation keyboard shortcuts

Expand table

Action	Shortcut
Move to the SQL Server Management Studio menu bar	ALT
Activate the menu for a tool component	ALT+HYPHEN
Display the context menu	SHIFT+F10
Display the **New File** dialog box to create a file	CTRL+N
Display the **New Project** dialog box to create a new project	CTRL+SHIFT+N
Display the **Open File** dialog box to open an existing file	CTRL+O or CTRL+SHIFT+G
Display the **Open Project** dialog box to open an existing project	CTRL+SHIFT+O
Display the **Add New Item** dialog box to add a new file to the current project	CTRL+SHIFT+A
Display the **Add Existing Item** dialog box to add an existing file to the current project	SHIFT+ALT+A

Action	Shortcut
Display the Query Designer	CTRL+SHIFT+Q
Close a menu or dialog box, canceling the action	ESC

Windows Management and Toolbar keyboard shortcuts

Expand table

Action	Shortcut
Close the current MDI child window	CTRL+F4
Close a menu or dialog box, cancels an operation in progress, or focuses on the current document window	ESC
Print	CTRL+P
Exit	ALT+F4
Toggle full screen mode	SHIFT+ALT+ENTER
Close the current tool window	SHIFT+ESC
Cycle through the next MDI child windows	CTRL+F6
Display the IDE navigator with the first document window selected	CTRL+TAB
Cycle through the previous MDI child windows	CTRL+SHIFT+TAB
Moves the insertion point to the dropdown bar located at the top of the code editor when the editor is in Code view or Server Code view	CTRL+F2
Move to the current tool window toolbar	SHIFT+ALT
Display the IDE navigator with the first tool window selected	ALT+F7
Move to the next tool window	ALT+F6 or F6 in the Database Engine Query Editor
Move to the previous tool window	SHIFT+ALT+F7

Action	Shortcut
Move to the next pane of a split pane view of a single document	F6
Move to the previously selected window	SHIFT+ALT+F6 or SHIFT+F6 in the Database Engine Query Editor
Move to the previous pane of a split pane view of a single document	SHIFT+F6
Display the dock menu	ALT+MINUS SIGN (-)
Display a popup listing all open windows	CTRL+ALT+DOWN ARROW
Opens a new query editor window	CTRL+O
Display Object Explorer	F8
Display Registered Servers	CTRL+ALT+G
Display Template Explorer	CTRL+ALT+T
Display Solution Explorer	CTRL+ALT+L
Display the Summary Window	F7
Display the Properties Window	F4
Display the **Output** window	CTRL+ALT+O
Display the **Task List** window	CTRL+\, T or CTRL+\, CTRL+T
Toggle between Object Explorer Details list view and Object Explorer Details property pane.	F6
Control the splitter bar that separates the Object Explorer Details list view and Object Explorer Details property pane to adjust the size of the display pane.	TAB, then UP arrow or DOWN arrow
Display the Toolbox	CTRL+ALT+X
Display the Bookmarks Window	CTRL+K, CTRL+W
Display the Browser Window	CTRL+ALT+R
Display a smart tag menu of common commands for Web server	SHIFT+ALT+F10

Action	Shortcut
controls in the HTML designer	
Display the Error List Window (Transact-SQL Editor only)	CTRL+\, CTRL+E or CTRL+\, E
Move to the next entry in the Error List window (Transact-SQL Editor only)	CTRL+SHIFT+F12
Display the previous page in the viewing history. Available only in the Web browser window	ALT+LEFT ARROW
Display the next page in the viewing history. Available only in the Web browser window	ALT+RIGHT ARROW

Cursor Movement keyboard shortcuts

Expand table

Action	Shortcut
Move the cursor left	LEFT ARROW
Move the cursor right	RIGHT ARROW
Move up the cursor	UP ARROW
Move the cursor down	DOWN ARROW
Move the cursor to the beginning of the line	HOME
Move the cursor to the end of the line	END
Move the cursor to the beginning of the document	CTRL+HOME
Move the cursor to the end of the document	CTRL+END
Move up the cursor one screen	PAGE UP
Move the cursor down one screen	PAGE DOWN
Moves the cursor one word to the right	CTRL+ RIGHT ARROW
Moves the cursor one word to the left	CTRL+ LEFT ARROW
Returns the cursor to the last item.	SHIFT+F8

Action	Shortcut
Moves the cursor to the top of the document	CTRL+PAGE UP
Moves to the previous tab in the document	CTRL+PAGE UP
Moves the cursor to the bottom of the document	CTRL+PAGE DOWN
Moves to the next tab in the document	CTRL+PAGE DOWN

Text Selection keyboard shortcuts

Expand table

Action	Shortcut
Select text from the cursor to the beginning of the document	CTRL+SHIFT+ HOME
Select text from the cursor to the end of the document	CTRL+SHIFT+END
Select text from the cursor to the start of the current line	SHIFT+HOME
Moves the cursor to the start of the current line and extends the column selection	SHIFT+ALT+HOME
Select text from the cursor to the end of the current line	SHIFT+END
Moves the cursor to the end of the line, extending the column selection.	SHIFT+ALT+END
Select text down line by line starting from the cursor	SHIFT+ DOWN ARROW
Moves the cursor down one line, extending the column selection	SHIFT+ Ctrl+Shift+Del
Moves the cursor one character to the left and extends the selection	SHIFT+LEFT ARROW
Moves the cursor one character to the left and extends the column selection	SHIFT+ALT+LEFT ARROW
Moves the cursor one character to the right and extends the selection	SHIFT+RIGHT ARROW
Moves the cursor one character to the right and extends the column selection	SHIFT+ALT+RIGHT ARROW
Select text up line by line starting from the cursor	SHIFT+UP ARROW
Move up the cursor one line, extending the selection	SHIFT+ALT+ UP ARROW

Action	Shortcut
Extend selection up one page	SHIFT+ PAGE UP
Extend selection down one page	SHIFT+ PAGE DOWN
Select the entire current document	CTRL+A
Select the word containing the cursor, or the closest word	CTRL+W
Select the current location in the editor, back to the previous location in the editor	CTRL+=
Extend the selection to the top of the current window	CTRL+SHIFT+ PAGE UP
Move the cursor to the last line in view, extending the selection	CTRL+SHIFT+ PAGE DOWN
Extend the selection one word to the right	CTRL+SHIFT+ RIGHT ARROW
Extend the selection one word to the left	CTRL+SHIFT+ LEFT ARROW
Move the cursor to the right one word, extending the selection	CTRL+SHIFT+ALT+ RIGHT ARROW
Move the cursor to the left one word, extending the selection	CTRL+SHIFT+ALT+ LEFT ARROW
Move the cursor to the next brace, extending the selection	CTRL+SHIFT+]
Select the text from the current location of the cursor to the Navigate Backward (CTRL+MINUS SIGN (-)) location	CTRL+EQUAL SIGN (=)
Go back to the previous document or window in the navigation history	CTRL+MINUS SIGN (-)
Go forward to the next document or window in the navigation history	CTRL+SHIFT+MINUS SIGN (-)
Swaps the anchor and end points of the current selection	CTRL+K, CTRL+A
Moves the cursor to the first line in view, extending the selection	CTRL+SHIFT+PAGE UP
Moves the cursor to the last line in view, extending the selection	CTRL+SHIFT+PAGE DOWN

Bookmark keyboard shortcuts

Expand table

Action	Shortcut
Set or remove a bookmark at the current line	CTRL+K, CTRL+K
Next bookmark	CTRL+K, CTRL+N
If the current bookmark is in a folder, moves to the next bookmark in the folder. Bookmarks outside the folder are skipped. If the bookmark isn't in a folder, moves to the next bookmark at the same level. If the Bookmarks window contains folder, bookmarks in folders are skipped.	CTRL+SHIFT+K, CTRL+SHIFT+N
Previous bookmark	CTRL+K, CTRL+P
If the current bookmark is in a folder, moves to the next bookmark in the folder. Bookmarks outside the folder are skipped. If the bookmark isn't in a folder, moves to the next bookmark at the same level. If the Bookmarks window contains folder, bookmarks in folders are skipped.	CTRL+SHIFT+K, CTRL+SHIFT+P
Clear bookmarks	CTRL+K, CTRL+L

Tree Control keyboard shortcuts

Expand table

Action	Shortcut
Collapse tree nodes	- (on the numeric keypad)
Expand all tree nodes	* (on the numeric keypad)
Scroll up the tree control in the window	CTRL+UP ARROW
Scroll the tree control down in the window	CTRL+DOWN ARROW

Code Editor keyboard shortcuts

All shortcuts aren't implemented in all types of code editors.

Expand table

Action	Shortcut
Toggle the full-screen display	SHIFT+ALT+ENTER
Scroll text up one line	CTRL+UP ARROW
Scroll text down one line	CTRL+DOWN ARROW
Reverse the last editing action	CTRL+Z or ALT+BACKSPACE
Restore the previously undone edit	CTRL+SHIFT+Z or CTRL+Y or ALT+SHIFT+BACKSPACE
Save the selected item	CTRL+S
Save all	CTRL+SHIFT+S
Close	CTRL+F4
Print	CTRL+P
Exit	ALT+F4
Open the current file in a browser	CTRL+SHIFT+W
Delete all text in the current file	CTRL+SHIFT+DEL
Display the **Go To Line** dialog box	CTRL+G
Display the **Navigate To** dialog box.	CTRL+PLUS SIGN (+)
Increase line indent	TAB
Decrease line indent	SHIFT+TAB
Make the selected text upper case	CTRL+SHIFT+U
Make the selected text lower case	CTRL+U
Make the selected text a comment	CTRL+K, CTRL+C
Uncomment the selected text	CTRL+K, CTRL + U
Open a new query with current connection	CTRL+N
Open database in Object Explorer	ALT+F8

Action	Shortcut
Specify values for template parameters	CTRL+SHIFT+M
Run the selected portion of the query editor or the entire query editor if nothing is selected	F5
Parse the selected portion of the query editor or the entire query editor if nothing is selected	CTRL+F5
Display the estimated execution plan	CTRL+SHIFT+ALT+L
Cancel the executing query	ALT+BREAK
Include actual execution plan in the query output	CTRL+SHIFT+ALT+M
Output results in a grid	CTRL+SHIFT+D
Output results in text format	CTRL+T
Output results to a file	CTRL+SHIFT+T
Show or hide the query results pane	CTRL+R
Show the query results pane	CTRL+SHIFT+ALT+R
Toggle between query and results pane	F6
Copy the result grid and headers to the clipboard	CTRL+SHIFT+C
Move to the next active window in SSMS	ALT+F6
Open SQL Server Profiler	CTRL+ALT+P
Display the Query Designer dialog from the query editor window	CTRL+SHIFT+Q
Run the **sp_help** system stored procedure	ALT+F1
Run the **sp_who** system stored procedure	CTRL+1
Run the **sp_lock** system stored procedure	CTRL+2
Run the stored procedure configured for this shortcut in the **Tools**, **Options**, **Keyboard**, **Query Shortcuts** dialog	CTRL+3
Run the stored procedure configured for this shortcut in the **Tools**, **Options**, **Keyboard**, **Query Shortcuts** dialog	CTRL+4
Run the stored procedure configured for this shortcut in the **Tools**, **Options**, **Keyboard**, **Query Shortcuts** dialog	CTRL+5
Run the stored procedure configured for this shortcut in the **Tools**, **Options**, **Keyboard**, **Query Shortcuts** dialog	CTRL+6

Action	Shortcut
Run the stored procedure configured for this shortcut in the **Tools**, **Options**, **Keyboard**, **Query Shortcuts** dialog	CTRL+7
Run the stored procedure configured for this shortcut in the **Tools**, **Options**, **Keyboard**, **Query Shortcuts** dialog	CTRL+7
Run the stored procedure configured for this shortcut in the **Tools**, **Options**, **Keyboard**, **Query Shortcuts** dialog	CTRL+8
Run the stored procedure configured for this shortcut in the **Tools**, **Options**, **Keyboard**, **Query Shortcuts** dialog	CTRL+9
Run the stored procedure configured for this shortcut in the **Tools**, **Options**, **Keyboard**, **Query Shortcuts** dialog	CTRL+0

Text Manipulation in Code Editor keyboard shortcuts

Expand table

Action	Shortcut
Insert a new line	ENTER or SHIFT+ENTER
Swap the characters on either side of the cursor (doesn't apply to the SQL Editor.)	CTRL+T
Delete one character to the right of the cursor	DELETE
Delete one character to the left of the cursor	BACKSPACE or SHIFT+ BACKSPACE
Delete whitespace in the selection, or deletes whitespace next to the cursor if there's no selection	CTRL+K, C
Insert the number of spaces configured for the editor	TAB
Insert a blank line above the cursor	CTRL+ENTER
Insert a blank line below the cursor	CTRL+SHIFT+ ENTER

Action	Shortcut
Change the selected text to lowercase	CTRL+SHIFT+L
Change the selected text to uppercase	CTRL+SHIFT+U
Toggle between insert mode and overtype mode	INSERT
Move selected lines to the left on tab stop	SHIFT+TAB
Delete the word to the right of the cursor	CTRL+DELETE
Delete the word to the left of the cursor	CTRL+BACKSPACE
Transpose the words on either side of the cursor (doesn't apply to the SQL Editor.)	
Moves the line containing the cursor below the next line	SHIFT+ALT+T
Applies the indenting and space formatting for the language specified on the **Formatting** pane of the language in the **Text Editor** section of the **Options** dialog. Available only in the text editor.	CTRL+K, CTRL+D
Correctly indents the selected lines of code based on the surrounding lines of code	CTRL+K, CTRL+F
Set or remove a shortcut in the current line	CTRL+K, CTRL+H
Remove the comment syntax from the current line	CTRL+K, CTRL+U
Shows or hides spaces and tabs	CTRL+R, CTRL+W
Enables or disables word wrap in an editor	Alt+F, CTRL+W
Collapses all outlining regions to show just the outermost groups in the hierarchy	CTRL+M, CTRL+A
Collapses the currently selected outlining region	CTRL+M, CTRL+S
Expands all outlining regions on the page	CTRL+M, CTRL+X
Expands the currently selected outlining region	CTRL+M, CTRL+E
Collapses existing outlining regions	CTRL+M, CTRL+O
Hides the selected text. A signal icon marks the location of the hidden text	CTRL+M, CTRL+H
Toggles all text sections previously marked as hidden between the hidden and display states.	CTRL+M, CTRL+L
Toggles the currently selected hidden text section between the hidden and display states	CTRL+M, CTRL+M

Action	Shortcut
Removes all outlining information in the document	CTRL+M, CTRL+P
Removes the outlining information for the currently selected region	CTRL+M, CTRL+U

Microsoft IntelliSense keyboard shortcuts

Expand table

Action	Shortcut
List members	CTRL+J
Complete word	CTRL+SPACE or ALT+RIGHT ARROW
Display quick information	CTRL+K, CTRL+I
Display parameter information	CTRL+SHIFT+SPACE
Copy parameter tip	CTRL+SHIFT+ALT+C
Paste parameter tip	CTRL+SHIFT+ALT+P
Jump between syntax pairs	CTRL+]
Launch code snippet picker	CTRL+K, CTRL+X
Refresh local cache	CTRL+SHIFT+R
Launch Surround With snippet picker	CTRL+K, CTRL+S
Display the Code Snippet Manager	CTRL+K, CTRL+B
Changes the IntelliSense filter level from the **Common** tab to the **All** tab.	ALT+PLUS SIGN (+)
Changes the IntelliSense filter level from the **All** tab to the **Common** tab.	ALT+PERIOD (.)

Document Window and Browser keyboard shortcuts

Expand table

Action	Shortcut
Toggle full-screen mode	SHIFT+ALT+ENTER
Move to the next pane of a split pane view of a document	F6
Move to the previous document in the editor or designer	CTRL+SHIFT+F6 CTRL+SHIFT+TAB
Move to the previous pane of a document in split pane view	SHIFT+F6
Back, display the previous page in the viewing history	ALT+LEFTARROW
Forward, display the next page in the viewing history	ALT+RIGHTARROW
Closes a menu or dialog box, cancels an operation in progress, or places focus in the current window	ESC

Solution Explorer keyboard shortcuts

Expand table

Action	Shortcut
Display Solution Explorer	CTRL+ALT+L
Display the **New File** dialog box to create a new file	CTRL+N
Display the **New Project** dialog box to create a new project	CTRL+SHIFT+N
Display the **Open File** dialog box to open an existing file	CTRL+O
Change the name of the selected object	F2

Help and Books Online keyboard shortcuts

Expand table

Action	Shortcut
Help	F1 or SHIFT+F1
Display SQL Server Books Online	CTRL+F1

Action	Shortcut
Open the Help Library Manager	CTRL+ALT+F1
Display the SQL Server Resource Center Web page	CTRL+ALT+F2
Display help for the current editor window	SHIFT+F1

Search keyboard shortcuts

Expand table

Action	Shortcut
Display the **Find** dialog box	CTRL+F
Displays the definition for the selected symbol.	F12
Displays the list of references for the selected symbol.	SHIFT+F12
Display the **Replace** dialog box	CTRL+H
Start incremental search. Type the characters to search for or press CTRL+I to search for characters from the previous search	CTRL+I
Find the next occurrence of the previous search text	F3
Find the previous occurrence of the search text	SHIFT+F3
Find the next occurrence of the currently selected text	CTRL+F3
Find the previous occurrence of the currently selected text	CTRL+SHIFT+F3
Display the **Replace in Files** dialog box	CTRL+SHIFT+H
Reverse incremental search so it starts at the bottom of the file and searches to the top	CTRL+SHIFT+I
Select or clear the **Search up** option in **Find and Replace**	ALT+F3, B
Stop the **Find in Files** search	ALT+F3, S
Select or clear the **Find whole word** option in **Find and Replace**	ALT+F3, W
Selects or clears the **Wildcard** option in **Find and Replace**	ALT+F3, P
Place the caret in the Find/Command box of the Standard toolbar	CTRL+/

Cut and Paste keyboard shortcuts

Expand table

Action	Shortcut
Cut (delete the currently selected item and place it to the Clipboard)	CTRL+X
Cuts all of the selected lines, or the current line if nothing is selected.	CTRL+L or CTRL+SHIFT+L
Copy to the Clipboard	CTRL+C or CTRL+INSERT
Paste from the Clipboard at the insertion point	CTRL+V or SHIFT+INSERT
Pastes an item from the Clipboard Ring at the insertion point and automatically selects the pasted item	CTRL+SHIFT+V

Activity Monitor keyboard shortcuts

Expand table

Action	Shortcut
Launches Activity Monitor	CTRL+ALT+A
Closes Activity Monitor	CTRL+F4
Refresh	F5
Filter the monitor display	CTRL+SHIFT+F
Cycle through panels	F6
Expand or collapse selected pane	CTRL and + or -
Expand or collapse all panes	+ or -
Copies entire selected row in grid	CTRL+C
Copy cell	CTRL+SHIFT+C
Dropdown for filtering in grid	ALT+DOWN

Action	Shortcut
Scroll up or down Activity Monitor	CTRL+ALT+UP/DOWN

Replication Monitor keyboard shortcuts

Expand table

Action	Shortcut
Refresh	F5
Open a detail window from a grid	ENTER

Replication Conflict Viewer keyboard shortcuts

Expand table

Action	Shortcut
Define filter	F6
Apply filter	F7
Show all columns	F8

Query Designer keyboard shortcuts

Expand table

Action	Shortcut
Cancels or stops the currently running query	CTRL+T
Displays the diagram pane of the **Query Designer**	CTRL+1
Displays the criteria pane of the **Query Designer**	CTRL+2
Displays the SQL pane of the **Query Designer**	CTRL+3
Displays the results pane of the **Query Designer**	CTRL+4
Run the query specified in the **Query Designer**	CTRL+R

Action	Shortcut
When in the results pane, moves focus to the tool strip docked at the bottom of the designer	CTRL+G
Enables JOIN mode in the **Query Designer**	CTRL+SHIFT+J

Designer keyboard shortcuts

Expand table

Action	Shortcut
Move the selected control down in increments of 8 on the design surface	DOWN ARROW
Move the selected control left in increments of 8 on the design surface	LEFT ARROW
Move the selected control right in increments of 8 on the design surface	RIGHT ARROW
Move up the selected control in increments of 8 on the design surface	UP ARROW
Increases the height of the selected control in increments of 8	SHIFT+DOWN ARROW
Reduces the width of the selected control in increments of 8	SHIFT+LEFT ARROW
Increases the width of the selected control in increments of 8	SHIFT+RIGHT ARROW
Decreases the height of the selected control in increments of 8	SHIFT+UP ARROW
Moves to the next control on the page	TAB
Moves to the previous control on the page	SHIFT+TAB
Display the grid on the design surface	ENTER

Related content

- Customize Menus and Shortcut Keys

Feedback

Was this page helpful? Yes No

Log File Viewer

Article • 02/28/2023

Applies to: SQL Server

Log File Viewer in SQL Server Management Studio is used to access information about errors and events that are captured in log files.

Benefits of using Log File Viewer

You can view SQL Server log files from a local or remote instance of SQL Server when the target instance is offline or cannot start. You can access the offline log files from Registered Servers, or programmatically through WMI and WQL (WMI Query Language) queries. For more information, see View Offline Log Files. Following are the types of log files you can access using Log File Viewer:

- Audit Collection
- Data Collection
- Database Mail
- Job History
- Maintenance Plans
- Remote Maintenance Plans
- SQL Server
- SQL Server Agent
- Windows NT (These are Windows events that can also be accessed from Event Viewer.)

Log File Viewer Tasks

Expand table

Task Description	Topic
Describes how to open Log File Viewer depending on the information that you want to view.	Open Log File Viewer
Describes how to view offline log files through registered servers and how to verify WMI permissions.	View Offline Log Files

Task Description	Topic
Provides Log File Viewer F1 Help.	Log File Viewer F1 Help

See Also

SQL Server Audit (Database Engine)
SQL Server Agent Error Log

Log File Viewer F1 Help

Article • 02/28/2023

Applies to: SQL Server

Log File Viewer displays log information from many different components. When Log File Viewer is open, use the **Select logs** pane to select the logs you want to display. Each log displays columns appropriate to that kind of log.

The logs that are available depend on how Log File Viewer is opened. For more information, see Open Log File Viewer.

The number of rows that are displayed for audit logs can be configured on the **SQL Server Object Explorer/Commands** page of the **Tools/Options** dialog box. For descriptions of the columns that are displayed for audit logs, see sys.fn_get_audit_file (Transact-SQL).

Options

Load Log
Open a dialog box where you can specify a log file to load.

Export
Open a dialog box that lets you export the information that is shown in the **Log file summary** grid to a text file.

Refresh
Refresh the view of the selected logs. The **Refresh** button rereads the selected logs from the target server while applying any filter settings.

Filter
Open a dialog box that lets you specify settings that are used to filter the log file, such as **Connection**, **Date**, or other **General** filter criteria.

Search
Search the log file for specific text. Searching with wildcard characters is not supported.

Stop
Stops loading the log file entries. For example, you can use this option if a remote or offline log file takes a long time to load, and you only want to view the most recent entries.

Log file summary
This information panel displays a summary of the log file filtering. If the file is not filtered, you will see the following text, **No filter applied**. If a filter is applied to the log, you will see the following text, **Filter log entries where:** <filter criteria>.

Selected row details
Select a row to display additional details about the selected event row at the bottom of the page. The columns can be reordered by dragging them to new locations in the grid. The columns can be resized by dragging the column separator bars in the grid header to the left or right. Double-click the column separator bars in the grid header to automatically size the column to the content width.

Instance
The name of the instance on which the event occurred. This is displayed as *computer name**instance name*.

Frequently Displayed Columns

Date
Displays the date of the event.

Source
Displays the source feature from which the event is created, such as the name of the service (MSSQLSERVER, for example). This does not appear for all log types.

Message
Displays any messages associated with the event.

Log Type
Displays the type of log to which the event belongs. All selected logs appear in the log file summary window.

Log Source
Displays a description of the source log in which the event is captured.

Permissions

To access log files for instances of SQL Server that are online, this requires membership in the securityadmin fixed server role.

To access log files for instances of SQL Server that are offline, you must have read access to both the **Root\Microsoft\SqlServer\ComputerManagement10** WMI namespace, and to the folder where the log files are stored. For more information, see the Security section of the topic View Offline Log Files.

See Also

Log File Viewer
Open Log File Viewer
View Offline Log Files

Open Log File Viewer

Article • 02/28/2023

Applies to: SQL Server

You can use Log File Viewer in SQL Server Management Studio to access information about errors and events that are captured in the following logs:

- Audit Collection
- Data Collection
- Database Mail
- Job History
- SQL Server
- SQL Server Agent
- Windows events (These Windows events can also be accessed from Event Viewer.)

Beginning in SQL Server 2012 (11.x), you can use Registered Servers to view SQL Server log files from local or remote instances of SQL Server. By using Registered Servers, you can view the log files when the instances are either online or offline. For more information about online access, see the procedure "To view online log files from Registered Servers" later in this topic. For more information about how to access offline SQL Server log files, see View Offline Log Files.

You can open Log File Viewer in several ways, depending on the information that you want to view.

Permissions

To access log files for instances of SQL Server that are online, this requires membership in the securityadmin fixed server role.

To access log files for instances of SQL Server that are offline, you must have read access to both the **Root\Microsoft\SqlServer\ComputerManagement10** WMI namespace, and to the folder where the log files are stored. For more information, see the Security section of the topic View Offline Log Files.

Security

Requires membership in the securityadmin fixed server role.

View Log Files

To view logs that are related to general SQL Server activity

1. In Object Explorer, expand **Management**.
2. Do either of the following:
   - Right-click **SQL Server Logs**, point to **View**, and then click either **SQL Server Log** or **SQL Server and Windows Log**.
   - Expand **SQL Server Logs**, right-click any log file, and then click **View SQL Server Log**. You can also double-click any log file.

   The logs include **Database Mail**, **SQL Server**, **SQL Server Agent**, and **Windows NT**.

To view logs that are related to jobs

- In Object Explorer, expand **SQL Server Agent**, right-click **Jobs**, and then click **View History**.

  The logs include **Database Mail**, **Job History**, and **SQL Server Agent**.

To view logs that are related to maintenance plans

- In Object Explorer, expand **Management**, right-click **Maintenance Plans**, and then click **View History**.

  The logs include **Database Mail**, **Job History**, **Maintenance Plans**, **Remote Maintenance Plans**, and **SQL Server Agent**.

To view logs that are related to Data Collection

- In Object Explorer, expand **Management**, right-click **Data Collection**, and then click **View Logs**.

  The logs include **Data Collection**, **Job History**, and **SQL Server Agent**.

To view logs that are related to Database Mail

- In Object Explorer, expand **Management**, right-click **Database Mail**, and then click **View Database Mail Log**.

The logs include **Database Mail, Job History**, **Maintenance Plans**, **Remote Maintenance Plans**, **SQL Server**, **SQL Server Agent**, and **Windows NT**.

To view logs that are related to audits collections

- In Object Explorer, expand **Security**, expand **Audits**, right-click an audit, and then click **View Audit Logs**.

  The logs include **Audit Collection** and **Windows NT**.

See Also

Log File Viewer
SQL Server Audit (Database Engine)
View Offline Log Files

View Offline Log Files

Article • 02/28/2023

Applies to: SQL Server

Beginning in SQL Server 2012 (11.x), you can view SQL Server log files from a local or remote instance of SQL Server when the target instance is offline or cannot start.

You can access the offline log files from Registered Servers, or programmatically through WMI and WQL (WMI Query Language) queries.

> **Note**
>
> You can also use these methods to connect to an instance that is online, but for some reason, you cannot connect through a SQL Server connection.

Before you Begin

To connect to offline log files, an instance of SQL Server must be installed on the computer that you are using to view the offline log files, and on the computer where the log files that you want to view are located. If an instance of SQL Server is installed on both computers, you can view offline files for instances of SQL Server, and for instances that are running earlier versions of SQL Server on either computer.

If you are using Registered Servers, the instance that you want to connect to must be registered under **Local Server Groups** or under **Central Management Servers**. (The instance can be registered on its own or be a member of a server group.) For more information about how to add an instance of SQL Server to Registered Servers, see the following topics:

- Create or Edit a Server Group (SQL Server Management Studio)
- Register a Connected Server (SQL Server Management Studio)
- Create a Central Management Server and Server Group (SQL Server Management Studio)

For more information about how to view offline log files programmatically through WMI and WQL queries, see the following topics:

- SqlErrorLogEvent Class (This topic shows how to retrieve values for logged events in a specified log file.)
- SqlErrorLogFile Class (This topic shows how to retrieve information about all SQL Server log files on a specified instance of SQL Server.)

Permissions

To connect to an offline log file, you must have the following permissions on both the local and remote computers:

- Read access to the **Root\Microsoft\SqlServer\ComputerManagement12** WMI namespace. By default, everyone has read access through the Enable Account permission. For more information, see the "To verify WMI permissions" procedure later in this section.
- Read permission to the folder that contains the error log files. By default the error log files are located in the following path (where <*Drive*> represents the drive where you installed SQL Server and <*InstanceName*> is the name of the instance of SQL Server):

  <Drive>:\Program Files\Microsoft SQL Server\MSSQL13.<InstanceName>\MSSQL\Log

To verify WMI namespace security settings, you can use the WMI Control snap-in.

To verify WMI permissions

1. Open the WMI Control snap-in. To do this, do either of the following, depending on the operating system:
   - Click **Start**, type **wmimgmt.msc** in the **Start Search** box, and then press ENTER.
   - Click **Start**, click **Run**, type **wmimgmt.msc**, and then press ENTER.
2. By default, the WMI Control snap-in manages the local computer.

   If you want to connect to a remote computer, follow these steps:

   a. Right-click **WMI Control (Local)**, and then click **Connect to another computer**.

   b. In the **Change managed computer** dialog box, click **Another computer**.

   c. Enter the remote computer name, and then click **OK**.
3. Right-click **WMI Control (Local)** or **WMI Control (***RemoteComputerName***)**, and then click **Properties**.
4. In the **WMI Control Properties** dialog box, click the **Security** tab.
5. In the namespace tree, locate and then click the following namespace:

   Root\Microsoft\SqlServer\ComputerManagement10
6. Click **Security**.

7. Make sure that the account that will be used has the **Enable Account** permission. This permission allows Read access to WMI objects.

View Log Files

The following procedure shows how to view offline log files through Registered Servers. The procedure assumes the following:

The instance of SQL Server that you want to connect to is already registered in Registered Servers.

To view log files for instances that are offline

1. If you want to view offline log files on a local instance, make sure that you start SQL Server Management Studio with elevated permissions. To do this, when you start Management Studio, right-click **SQL Server Management Studio**, and then click **Run as administrator**.

2. In SQL Server Management Studio, on the **View** menu, click **Registered Servers**.

3. In the console tree, locate the instance on which you want to view the offline files.

4. Do one of the following:

   - If the instance is under **Local Server Groups**, expand **Local Server Groups**, expand the server group (if the instance is a member of a group), right-click the instance, and then click **View SQL Server Log**.

   - If the instance is the Central Management Server itself, expand **Central Management Servers**, right-click the instance, point to **Central Management Server Actions**, and then click **View SQL Server Log**.

   - If the instance is under **Central Management Servers**, expand **Central Management Servers**, expand the Central Management Server, right-click the instance (or expand a server group and right-click the instance), and then click **View SQL Server Log**.

5. If you are connecting to a local instance, the connection is made using the current user credentials.

   If you are connecting to a remote instance, in the **Log File Viewer - Connect As** dialog box, do either of the following:

   - To connect as the current user, make sure that the **Connect as another user** check box is cleared, and then click **OK**.

- To connect as another user, select the **Connect as another user** check box, and then click **Set User**. When you are prompted, enter the user credentials (with the user name in the format *domain_name**user_name*), click **OK**, and then click **OK** again to connect.

> **Note**
>
> If the log files take too long to load, you can click **Stop** on the Log File Viewer toolbar.

See Also

Log File Viewer

SQL Server Agent

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

This article provides an overview of the SQL Server Agent, which is a Microsoft Windows service that executes scheduled administrative tasks (called *jobs*) in SQL Server and Azure SQL Managed Instance.

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Benefits of SQL Server Agent

SQL Server Agent uses SQL Server to store job information. Jobs contain one or more job steps. Each step contains its own task, for example, backing up a database.

SQL Server Agent can run a job on a schedule, in response to a specific event, or on demand. For example, if you want to back up all the company servers every weekday after hours, you can automate this task. Schedule the backup to run after 22:00 Monday through Friday. If the backup encounters a problem, SQL Server Agent can record the event and notify you.

> **Note**
>
> By default, the SQL Server Agent service is disabled when SQL Server is installed unless the user explicitly chooses to autostart the service.

SQL Server Agent Components

SQL Server Agent uses the following components to define the tasks to be performed, when to perform the tasks, and how to report the success or failure of the tasks.

Use the SQL Server Configuration Manager to manage the SQL Server Agent service, and use SQL Server Management Studio (SSMS) to easily manage SQL Server Agent properties, jobs, alerts, operators, and proxies in a graphical user interface.

Jobs

A *job* is a specified series of actions that SQL Server Agent performs. Use jobs to define an administrative task that can be run one or more times and monitored for success or failure. A job can run on one local server or on multiple remote servers.

> **Important**
>
> SQL Server Agent jobs that are running at the time of a failover event on a SQL Server failover cluster instance do not resume after failover to another failover cluster node. SQL Server Agent jobs that are running at the time a Hyper-V node is paused do not resume if the pause causes a failover to another node. Jobs that begin but fail to complete because of a failover event are logged as started, but do not show additional log entries for completion or failure. SQL Server Agent jobs in these scenarios appear to have never ended.

You can run jobs in several ways:

- According to one or more schedules.
- In response to one or more alerts.
- By executing the `sp_start_job` stored procedure.

Each action in a job is a *job step*. For example, a job step might consist of running a Transact-SQL statement, executing an SSIS package, or issuing a command to an Analysis Services server. Job steps are managed as part of a job.

Each job step runs in a specific security context. For job steps that use Transact-SQL, use the EXECUTE AS statement to set the security context for the job step. For other types of job steps, use a proxy account to set the security context for the job step.

Use the sp_help_job system stored procedure to find out information about a specific job. Use the dbo.sysjobs system table to view information about jobs. For example, use the following Transact-SQL (T-SQL) statement to view information about all jobs on a server:

```SQL
USE MSDB
GO
SELECT job_id, [name] FROM dbo.sysjobs;
```

Schedules

A *schedule* specifies when a job runs. More than one job can run on the same schedule, and more than one schedule can apply to the same job. A schedule can define the following conditions for the time when a job runs:

- Whenever SQL Server Agent starts.
- Whenever CPU utilization of the computer is at a level you've defined as idle.
- One time, at a specific date and time.
- On a recurring schedule.

For more information, see Create and attach schedules to jobs.

Alerts

An *alert* is an automatic response to a specific event. For example, an event can be a job that starts or system resources that reach a specific threshold. You define the conditions under which an alert occurs.

An alert can respond to one of the following conditions:

- SQL Server events
- SQL Server performance conditions
- Microsoft Windows Management Instrumentation (WMI) events on the computer where SQL Server Agent is running

An alert can perform the following actions:

- Notify one or more operators
- Run a job

For more information, see Alerts.

Operators

An *operator* defines contact information for an individual responsible for the maintenance of one or more instances of SQL Server. In some enterprises, operator responsibilities are assigned to one individual. In enterprises with multiple servers, many

individuals can share operator responsibilities. An operator doesn't contain security information, and doesn't define a security principal.

SQL Server can notify operators of alerts through...

- E-mail
- Pager (through e-mail)
- **net send**

> ⓘ **Note**
>
> To send notifications by using **net send**, the Windows Messenger service must be started on the computer where SQL Server Agent resides.

> ⓘ **Important**
>
> The Pager and **net send** options is removed from SQL Server Agent in a future version of SQL Server. Avoid using these features in new development work, and plan to modify applications that currently use these features.

To send notifications to operators by using e-mail or pagers, you must configure SQL Server Agent to use Database Mail. For more information, see Database Mail.

You can define an operator as the alias for a group of individuals. In this way, all members of that alias aren't verified at the same time. For more information, see Operators.

Security for SQL Server Agent administration

SQL Server Agent uses the **SQLAgentUserRole**, **SQLAgentReaderRole**, and **SQLAgentOperatorRole** fixed database roles in the `msdb` database to control access to SQL Server Agent for users who aren't members of the **sysadmin** fixed server role. In addition to these fixed database roles, subsystems and proxies help database administrators ensure that each job step runs with the minimum permissions required to perform its task.

Roles

Members of the **SQLAgentUserRole**, **SQLAgentReaderRole**, and **SQLAgentOperatorRole** fixed database roles in `msdb`, and members of the **sysadmin** fixed server role have access to SQL Server Agent. A user that doesn't belong to any of these roles can't use SQL Server Agent. For more information on the roles used by SQL Server Agent, see Implement SQL Server Agent Security.

Subsystems

A subsystem is a predefined object that represents functionality that is available to a job step. Each proxy has access to one or more subsystems. Subsystems provide security because they delimit access to the functionality that is available to a proxy. Each job step runs in the context of a proxy, except for Transact-SQL job steps. Transact-SQL job steps use the EXECUTE AS command to set the security context to the owner of the Job.

SQL Server defines the subsystems listed in the following table:

Expand table

Subsystem name	Description
Microsoft ActiveX Script	Run an ActiveX scripting job step. **Warning** The ActiveX Scripting subsystem is removed from SQL Server Agent in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature.
Operating System (**CmdExec**)	Run an executable program.
PowerShell	Run a PowerShell scripting job step.
Replication Distributor	Run a job step that activates the replication Distribution Agent.
Replication Merge	Run a job step that activates the replication Merge Agent.
Replication Queue Reader	Run a job step that activates the replication Queue Reader Agent.
Replication Snapshot	Run a job step that activates the replication Snapshot Agent.
Replication Transaction Log Reader	Run a job step that activates the replication Log Reader Agent.

Subsystem name	Description
Analysis Services Command	Run an Analysis Services command.
Analysis Services Query	Run an Analysis Services query.
SSIS package execution	Run an SSIS package.

> **Note**
>
> Because Transact-SQL job steps do not use proxies, there is no SQL Server Agent subsystem for Transact-SQL job steps.

SQL Server Agent enforces subsystem restrictions even when the security principal for the proxy would normally have permission to run the task in the job step. For example, a proxy for a user that is a member of the sysadmin fixed server role can't run an SSIS job step unless the proxy has access to the SSIS subsystem, even though the user can run SSIS packages.

Proxies

SQL Server Agent uses proxies to manage security contexts. A proxy can be used in more than one job step. Members of the **sysadmin** fixed server role can create proxies.

Each proxy corresponds to a security credential. Each proxy can be associated with a set of subsystems and a set of logins. The proxy can be used only for job steps that use a subsystem associated with the proxy. To create a job step that uses a specific proxy, the job owner must either use a login associated with that proxy or a member of a role with unrestricted access to proxies. Members of the **sysadmin** fixed server role have unrestricted access to proxies. Members of **SQLAgentUserRole**, **SQLAgentReaderRole**, or **SQLAgentOperatorRole** can only use proxies to which they have been granted specific access. Each user that is a member of any of these SQL Server Agent fixed database roles must be granted access to specific proxies so that the user can create job steps that use those proxies.

Automate administration

Use the following steps to configure SQL Server Agent to automate SQL Server administration:

1. Establish which administrative tasks or server events occur regularly and whether these tasks or events can be administered programmatically. A task is a good candidate for automation if it involves a predictable sequence of steps and occurs at a specific time or in response to a specific event.

2. Define a set of jobs, schedules, alerts, and operators by using SQL Server Management Studio, Transact-SQL scripts, or SQL Server Management Objects (SMO). For more information, see Create Jobs.

3. Run the SQL Server Agent jobs you've defined.

> **Note**
>
> For the default instance of SQL Server, the SQL Server service is named SQLSERVERAGENT. For named instances, the SQL Server Agent service is named SQLAgent$*instancename*.

If you're running multiple instances of SQL Server, you can use multiserver administration to automate tasks common across all instances. For more information, see Automated Administration Across an Enterprise.

Use the following tasks to get started with SQL Server Agent:

Expand table

Description	Article
Describes how to configure SQL Server Agent.	Configure SQL Server Agent
Describes how to start, stop, and pause the SQL Server Agent service.	Start, Stop, or Pause the SQL Server Agent Service
Describes considerations for specifying an account for the SQL Server Agent service.	Select an Account for the SQL Server Agent Service
Describes how to use the SQL Server Agent error log.	SQL Server Agent Error Log
Describes how to use performance objects.	Use Performance Objects
Describes the Maintenance Plan Wizard, which is a utility that you use create jobs, alerts, and operators to automate administration of an instance of SQL Server.	Use the Maintenance Plan Wizard
Describes how to automate administrative tasks using SQL Server Agent.	Automated Administration Tasks (SQL Server Agent)

NOSQLPS

Starting with SQL Server 2019, you can disable SQLPS. On the first line of a job step of the type PowerShell you can add `#NOSQLPS`, which stops the SQL Agent from auto-loading the SQLPS module. Now your SQL Agent Job runs the version of PowerShell installed on the machine, and then you can use any other PowerShell module you like.

To use the **SqlServer module** in your SQL Agent Job step, you can place this code on the first two lines of your script.

PowerShell

```
#NOSQLPS
Import-Module -Name SqlServer
```

Related content

- Surface area configuration
- SQL Server Agent with PowerShell

Feedback

Was this page helpful? Yes No

Post Download Instructions

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to specify download instructions for a target server.

Options

Instruction type
Specify the type of download instruction to post.

Description
View a description of the download instruction.

Polling interval
Set the polling interval for the target server. Applies only to the **Set Polling Interval** instruction.

All target servers
Select this option to post the download instruction to all target servers.

These target servers
Select this option to post the download instruction to the selected target servers.

Select
Specify that the target server should receive the download instruction.

Target Server
View the name of the target server.

Local time
View the date and time of the target server in the local time zone for that server.

Polling interval
View the current polling interval for the target server.

See Also

Automated Administration Across an Enterprise

Feedback

Was this page helpful? Yes No

Job Activity Monitor

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view the current activity of SQL Server Agent jobs. Click **Filter** to limit the jobs displayed. The **Agent Job Activity** grid is read-only. Select the column headers to sort the grid. To modify a job, double-click the job to open the **Job Properties** dialog box. Right-click a job in the grid to start it running all job steps, start at a particular job step, disable or enable the job, refresh the job, delete the job, view the history of the job, or view the properties of the job. Click **Refresh** to update the grid with current information.

Options

Name
Name of the job.

Enabled
Whether the job is enabled (**yes**) or not enabled (**no**).

Status*
Current status of the job.

Last Run Outcome
Job status when last run.

Last Run
Date and time job was last run using the local date and time of the server.

Next Run*
Date and time the job is next scheduled to run using the local date and time of the server.

Category
The job category assigned to the job.

Runnable
Yes if the job can be run; **No** if the job cannot be run. A job is not runnable if it has no steps or if it has no target server.

Scheduled
Yes if the job is assigned to a job schedule; **No** if the job has no schedule.

*Only members of the SQL Server sysadmin fixed server role and the server administrators group can see values in this column. Members of the SQLAgentOperatorRole role cannot see values in this column.

To open the Job Activity Monitor

- In **Object Explorer**, expand your server, expand **SQL Server Agent**, right-click **Job Activity Monitor**, and then click **View Job Activity**.

See Also

Monitor Job Activity

Feedback

Was this page helpful? Yes No

Automated Administration Tasks (SQL Server Agent)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Microsoft SQL Server allows you to automate administrative tasks. To automate administration, you define predictable administrative tasks and then specify the conditions under which each task occurs. Using automated administration to handle routine tasks and events frees your time to perform other administrative functions.

In This Section

Implement SQL Server Agent Security

Implement Jobs

Monitor and Respond to Events

Automated Administration Across an Enterprise

Feedback

Was this page helpful? Yes No

Implement SQL Server Agent Security

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

SQL Server Agent lets the database administrator run each job step in a security context that has only the permissions required to perform that job step, which is determined by a SQL Server Agent proxy. To set the permissions for a particular job step, you create a proxy that has the required permissions and then assign that proxy to the job step. A proxy can be specified for more than one job step. For job steps that require the same permissions, you use the same proxy.

The following section explains what database role you must grant to users so they can create or execute jobs by using SQL Server Agent.

Granting Access to SQL Server Agent

To use SQL Server Agent, users must be a member of one or more of the following fixed database roles:

- **SQLAgentUserRole**
- **SQLAgentReaderRole**
- **SQLAgentOperatorRole**

These roles are stored in the **msdb** database. By default, no user is a member of these database roles. Membership in these roles must be granted explicitly. Users who are members of the **sysadmin** fixed server role have full access to SQL Server Agent, and don't need to be a member of these fixed database roles to use SQL Server Agent. If a user isn't a member of one of these database roles or of the **sysadmin** role, the SQL Server Agent node isn't available to them when they connect to SQL Server by using SQL Server Management Studio.

Members of these database roles can view and execute jobs that they own, and create job steps that run as an existing proxy account. For more information about the specific permissions that are associated with each of these roles, see SQL Server Agent Fixed Database Roles.

Members of the **sysadmin** fixed server role have permission to create, modify, and delete proxy accounts. Members of the **sysadmin** role have permission to create job steps that don't specify a proxy, but instead run as the SQL Server Agent service account, which is the account that is used to start SQL Server Agent.

Guidelines

Follow these guidelines to improve the security of your SQL Server Agent implementation:

- Create dedicated user accounts specifically for proxies, and only use these proxy user accounts for running job steps.
- Only grant the necessary permissions to proxy user accounts. Grant only those permissions required to run the job steps that are assigned to a given proxy account.
- Don't run the SQL Server Agent service under a Microsoft Windows account that is a member of the Windows **Administrators** group.
- Proxies are only as secure as the SQL Server credential store.
- If user write operations can write to the NT Event log, they can raise alerts via SQL Server Agent.
- Don't specify the NT Admin account as a service account or proxy account.
- SQL Server and SQL Server Agent have access to each other's assets. The two services share a single process space and SQL Server Agent is a sysadmin on the SQL Server service.
- When a TSX (target server) enlists with an MSX (master server), the MSX sysadmins gets total control over the TSX instance of SQL Server.
- ACE is an extension and can't invoke itself. Chainer ScenarioEngine.exe (also known as Microsoft.SqlServer.Chainer.Setup.exe) can invoke ACE. Other host processes can also invoke ACE.

- ACE depends on the following configuration DLLs owned by SSDP, because those APIs of DLLs are called by ACE:
  - **SCO** - Microsoft.SqlServer.Configuration.Sco.dll, including new SCO validations for virtual accounts
  - **Cluster** - Microsoft.SqlServer.Configuration.Cluster.dll
  - **SFC** - Microsoft.SqlServer.Configuration.SqlConfigBase.dll
  - **Extension** - Microsoft.SqlServer.Configuration.ConfigExtension.dll

Linked servers

In some scenarios, such as with Azure SQL Managed Instance, to run a SQL Agent job that executes a Transact-SQL (T-SQL) query on a remote server through a linked server, you need to map a local login to a login on the remote server.

Use sp_addlinkedsrvlogin to create a mapping between a login on the local server to a login on the remote server that has the necessary permission to execute the T-SQL query. When the SQL Agent job connects to the remote server through the linked server, it executes the T-SQL query in the context of the remote login.

The following table describes how to map the login based on the SQL Agent job owner in Azure SQL Managed Instance:

Expand table

SQL Agent job owner	How to map the login
User that is not **sysadmin**	Map the local user that owns the SQL Agent job to the remote login.
sysadmin	Map all local users to the remote login by setting the `@locallogin` parameter to `NULL`.

> **Note**
>
> Creating logins on the remote server for SQL Agent jobs is required when the local server is Azure SQL Managed Instance. Failing to map users correctly can result in errors such as the following examples:
>
> - `Windows logins are not supported in this version of SQL Server`

- `Linked servers cannot be used under impersonation without a mapping for the impersonated login`

Related content

- Using Predefined Roles
- sp_addrolemember (Transact-SQL)
- sp_droprolemember (Transact-SQL)
- Security and Protection

Feedback

Was this page helpful? Yes No

Implement Jobs

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

You can use SQL Server Agent jobs to automate routine administrative tasks and run them on a recurring basis, making administration more efficient.

A job is a specified series of operations performed sequentially by SQL Server Agent. A job can perform a wide range of activities, including running Transact-SQL scripts, command-line applications, Microsoft ActiveX scripts, Integration Services packages, Analysis Services commands and queries, or Replication tasks. Jobs can run repetitive tasks or those that can be scheduled, and they can automatically notify users of job status by generating alerts, thereby greatly simplifying SQL Server administration.

You can run a job manually, or you can configure it to run according to a schedule or in response to alerts.

Related Tasks

Expand table

Description	Topic
Contains information about creating jobs and assigning ownership.	Create Jobs
Contains information about organizing jobs into categories.	Organize Jobs
Contains information about the different kinds of job steps you can create and how to manage them.	Manage Job Steps
Contains information about how to define when jobs start running and how often they should run.	Create and Attach Schedules to Jobs
Contains information about manually running jobs (without a schedule).	Run Jobs

Description	Topic
Contains information about how you can configure SQL Server Agent to respond to jobs. For example, you can configure SQL Server Agent to notify administrators when jobs are finished.	Specify Job Responses
Contains information about how to view existing jobs, their history once executes, and how to modify them.	View or Modify Jobs
Contains information about how to delete jobs.	Delete Jobs

See Also

Implement SQL Server Agent Security

Feedback

Was this page helpful? Yes No

Monitor and Respond to Events

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

SQL Server Agent can monitor and automatically respond to *events*, such as messages from SQL Server, specific performance conditions, and Windows Management Instrumentation (WMI) events.

In This Section

Alerts
Contains information about naming an alert and selecting the events or performance conditions to which alerts respond.

Create a User-Defined Event
Contains information about how to create events other than those that are predefined by SQL Server.

Operators
Contains information about creating aliases for administrators that SQL Server Agent can use to send notifications when jobs fail or succeed.

About Monitoring and Responding to Events

Automated responses to events are called *alerts*. You can define an alert on one or more events to specify how you want SQL Server Agent to respond to their occurrence. An alert can respond to an event by notifying an administrator or running a job, or both. An alert can also forward an event to the Microsoft Windows application log on a different computer. For example, you can specify that an operator be notified immediately if an event of severity 19 occurs. By defining alerts, database administrators can more effectively monitor and manage SQL Server.

SQL Server Agent only responds to events for which an alert is defined. The method that SQL Server Agent uses to monitor events depends on the type of event.

When a SQL Server Agent alert is defined for a performance counter, SQL Server Agent directly monitors the performance counter. For a WMI event, SQL Server Agent registers an event query for the WMI event.

To respond to messages from SQL Server, SQL Server Agent monitors the Windows application log. SQL Server Agent can only respond to messages that appear in this log. By default, SQL Server logs the following messages in the Windows application log:

- Severity 19 or higher sysmessages errors.

  If you also want to log specific sysmessages errors that have a severity lower than 19, use the sp_altermessage stored procedure to designate such errors as "always logged".

- Any RAISERROR statement invoked by using the WITH LOG syntax.

  Using RAISERROR WITH LOG is the recommended way to write to the Windows application log from an instance of SQL Server.

- Any application event that is logged by using xp_logevent.

  > **Note**
  >
  > Logging application events consumes log space and can cause the Windows application log to exceed its maximum size. Make sure that the maximum Windows application log size is large enough to avoid loss of SQL Server event information.

When SQL Server logs a message, the SQL Server Agent service compares the message against the alerts defined by the SQL Server administrator.

Regardless of the source of the event, the SQL Server Agent service responds to the event by performing the tasks specified in the alert for the event.

See Also

sp_altermessage

Feedback

Was this page helpful? Yes No

Automated Administration Across an Enterprise

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

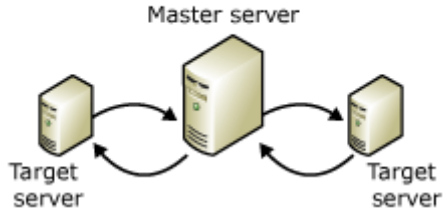
Automating administration across multiple instances of SQL Server is called *multiserver administration*. Use multiserver administration to do the following:

- Manage two or more servers.
- Schedule information flows between enterprise servers for data warehousing.

To take advantage of multiserver administration, you must have at least one master server and at least one target server. A master server distributes jobs to, and receives events from, target servers. A master server also stores the central copy of job definitions for jobs that are run on target servers. Target servers connect periodically to the master server to update their schedule of jobs. If a new job exists on the master server, the target server downloads the job. After the target server completes the job, it reconnects to the master server and reports the status of the job. Note that your job definition must be same when performing any database related activities.

The following illustration shows the relationship between master and target servers:

Multiserver administration configuration



If you administer departmental servers across a large corporation, you can define the following:

- One backup job with job steps.

- Operators to notify in case of backup failure.
- An execution schedule for the backup job.

Write this backup job one time on the master server and then enlist each departmental server as a target server. From the time of their enlistment, all the departmental servers run the same backup job, yet you defined the job only once.

> **Note**
>
> Multiserver administration features are intended for members of the sysadmin role. However, a member of the sysadmin role on the target server cannot edit the operations that are performed on the target server by the master server. This security measure prevents job steps from being accidentally deleted and operations on the target server from being interrupted.

In This Section

Create a Multiserver Environment
Contains information about how to create and manage master and target servers.

Choose the Right SQL Server Agent Service Account for Multiserver Environments
Contains information about how using nonadministrative Windows accounts or the Local System account for the SQL Server Agent service can affect multiserver environments.

Set Encryption Options on Target Servers
Contains information about setting the MsxEncryptChannelOptions SQL Server Agent registry subkey on target servers.

Manage Jobs Across an Enterprise
Contains information about checking job status, changing target servers for jobs, synchronizing target server clocks, and polling master servers for their current job status.

Troubleshoot Multiserver Jobs That Use Proxies
Contains information about troubleshooting multiserver jobs that use proxies which fail.

Poll Servers
Contains information about how to implicitly and explicitly make target servers poll master servers to synchronize jobs information.

Manage Events
Contains information about event forwarding from target servers to master servers.

Tune Automated Administration Across an Enterprise
Contains information about how automated administration in a multiserver environment takes advantage of the self-tuning features of SQL Server.

See Also

Backward Compatibility Topics for Installing the SQL Server Database Engine
Registering Servers
sp_add_targetservergroup
sp_delete_targetserver
sp_delete_targetservergroup
sp_help_downloadlist
sp_help_jobserver
sp_help_targetservergroup
sp_resync_targetserver
sp_update_targetservergroup
sysjobservers
syslogins
systargetservers

Feedback

Was this page helpful? Yes No

Change Steps of a SQL Server Agent Master Job

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to make changes to the steps of a SQL Server Agent master job in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Limitations and Restrictions

A SQL Server Agent master job cannot be targeted at both local and remote servers.

Security

Permissions

Unless you are a member of the **sysadmin** fixed server role, you can only modify jobs that you own. For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To make changes to the steps of a SQL Server Agent master job

1. In **Object Explorer,** click the plus sign to expand the server that contains the job where you want to modify steps.
2. Click the plus sign to expand **SQL Server Agent**.

3. Click the plus sign to expand the **Jobs** folder.
4. Right-click the job where you want to modify steps and select **Properties**.
5. In the **Job Properties** *-job_name* dialog box, under **Select a page**, select **Steps**.
6. Click **Edit** to open the **Job Step Properties** *-job_step_name* dialog box. For more information on the available options in this dialog box, see Job Step Properties - New Job Step (General Page) and Job Step Properties - New Job Step (Advanced Page).
7. When finished, click **OK**.
8. In the **Job Properties** *-job_name* dialog box, click **OK**.

Using Transact-SQL

To make changes to the steps of a SQL Server Agent master job

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- changes the number of retry attempts for the first step
-- of the Weekly Sales Data Backup job.
-- After running this example, the number of retry attempts is 10
USE msdb ;
GO

EXEC dbo.sp_update_jobstep
    @job_name = N'Weekly Sales Data Backup',
    @step_id = 1,
    @retry_attempts = 10 ;
GO
```

For more information, see sp_update_jobstep (Transact-SQL).

Feedback

Was this page helpful? Yes No

Change the Scheduling Details for a SQL Server Agent Master Job

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to change the scheduling details for a job definition in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Limitations and Restrictions

A SQL Server Agent master job cannot be targeted at both local and remote servers.

> **Warning**
>
> It is possible to attach multiple jobs to the same schedule. Note that updating a schedule will impact all the jobs attached to the schedule.

Security

Permissions

Unless you are a member of the **sysadmin** fixed server role, you can only modify jobs that you own. For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To change the scheduling details for a job definition

1. In **Object Explorer,** click the plus sign to expand the server that contains the job whose schedule you want to edit.
2. Click the plus sign to expand **SQL Server Agent**.
3. Click the plus sign to expand the **Jobs** folder.
4. Right-click the job whose schedule you want to edit and select **Properties**.
5. In the **Job Properties** *-job_name* dialog box, under **Select a page**, select **Schedules**. For more information on the available options on this page, see Job Properties - New Job (Schedules Page).
6. When finished, click **OK**.

Using Transact-SQL

To change the scheduling details for a job definition

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- changes the enabled status of the NightlyJobs schedule to 0
-- and sets the owner to terrid.
USE msdb ;
GO

EXEC dbo.sp_update_schedule
    @name = 'NightlyJobs',
    @enabled = 0,
    @owner_login_name = 'terrid' ;
GO
```

For more information, see sp_update_schedule (Transact-SQL).

Feedback

Was this page helpful? Yes No

SQL Server Agent F1 Help

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This section contains the F1 Help for SQL Server Agent. These topics are available from the user interface by pressing the F1 key or by selecting Help in dialog boxes.

Feedback

Was this page helpful? Yes No

Jobs Node (SQL Server Agent F1 Help)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This section contains the F1 Help for the **Jobs** node of Object Explorer in SQL Server Management Studio.

Feedback

Was this page helpful? Yes No

Alerts Node (SQL Server Agent F1 Help)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This section contains the F1 Help for the **Alerts** node of Object Explorer in SQL Server Management Studio.

Feedback

Was this page helpful? Yes No

Proxies Node (SQL Server Agent F1 Help)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This section contains the F1 Help for the **Proxies** node of Object Explorer in SQL Server Management Studio.

Feedback

Was this page helpful? Yes No

Operators Node (SQL Server Agent F1 Help)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This section contains the F1 Help for the **Operators** node of Object Explorer in SQL Server Management Studio.

Feedback

Was this page helpful? Yes No

Configure SQL Server Agent Error Logs (General Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this screen to view and update settings for SQL Server Agent error logging.

Options

Error log file
Specifies the file to which SQL Server Agent writes error logs.

...
Browse to the error log file.

Write OEM error log
Writes the error log file as a non-Unicode file. This reduces the amount of disk space consumed by the log file. However, messages that include Unicode data may be more difficult to read when this option is enabled.

Errors
Writes only errors and informational messages to the log file.

Warnings
Writes only warnings and informational messages to the log file.

Information
Writes only informational messages to the log file.

See Also

SQL Server Agent Error Log

Feedback

Was this page helpful? Yes No

Recycle SQL Server Agent Error Logs

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to recycle the Microsoft SQL Server Agent Error Logs. Recycling the log closes the current SQL Server Agent error log and starts a new error log without restarting the SQL Server Agent service. Notice that SQL Server Agent maintains the nine most recent error logs. If there are already nine error logs present, recycling the error log causes SQL Server Agent to remove the oldest error log.

See Also

SQL Server Agent Error Log

Feedback

Was this page helpful? Yes No

SQL Server Agent Properties (Advanced Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view and modify advanced properties of the Microsoft SQL Server Agent service.

Options

SQL Server event forwarding
The options in this category activate and configure event forwarding.

Forward events to a different server
Forwards SQL Server Agent events to a different server.

Server
Select the name of the server to forward events to.

Unhandled events
Forwards only unhandled events to the specified server. SQL Server Agent forwards only events that no alert responds to.

All events
Forwards all events. When an alert in the local instance responds to the event, SQL Server agent will both forward the event and process the alert.

If event has severity at or above
Forwards only events with a severity level at or above the level specified.

Idle CPU Condition
The options in this category define the conditions under which SQL Server Agent runs jobs scheduled to run on the Idle CPU schedule.

Define idle CPU condition

Defines the conditions under which SQL Server Agent considers the CPU to be idle.

Average CPU usage falls below

Percentage of CPU usage below which the CPU is considered to be idle.

And remains below this level for

Amount of time that the average CPU must below the level specified before SQL Server Agent runs jobs on the Idle CPU schedule.

See Also

Create and Attach Schedules to Jobs
Manage Events

Feedback

Was this page helpful? Yes No

SQL Server Agent Properties (Alert System Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view and modify the settings for messages sent by Microsoft SQL Server Agent alerts.

Options

Mail session
The options in this section configure SQL Server Agent Mail.

Enable mail profile
Enables SQL Server Agent Mail. By default, SQL Server Agent Mail is not enabled.

Mail System
Sets the mail system for SQL Server Agent to use. Database Mail

> **Note**
>
> After changing the e-mail system, you must restart the SQL Server Agent service for the change to take effect.

Mail Profile
Sets the profile for SQL Server Agent to use. You may also select **<new Database Mail profile...>** to create a new profile.

Pager e-mails
The options in this section allow you to configure e-mail messages sent to pager addresses to work with your paging system.

Address formatting for pager e-mails
This section allows you to specify the format of the addresses and the subject line included in pager e-mails.

To line
Specifies the options for the **To** line of the message

Prefix
Type any fixed text that your system requires at the beginning of the **To** line of messages sent to a pager.

Pager
Includes the e-mail address for the message between the prefix and the suffix.

Suffix
Type any fixed text that your paging system requires at the end of the **To** line of messages sent to a pager.

Cc line
Specifies options for the **Cc** line of the message.

Prefix
Type any fixed text that your system requires at the beginning of the **Cc** line of messages sent to a pager.

Pager
Includes the e-mail address for the message between the prefix and the suffix.

Suffix
Type any fixed text that your paging system requires at the end of the **Cc** line of messages sent to a pager.

Subject
Specifies the options for the subject of the message.

Prefix
Type any fixed text that your paging system requires at the beginning of the **Subject** line of messages sent to a pager.

Suffix
Type any fixed text that your paging system requires at the end of the **Subject** line of messages sent to a pager.

Include body of e-mail in notification message
Includes the body of the e-mail message in the message sent to the pager.

Fail-safe operator
This section allows you to specify the options for the fail-safe operator.

Enable fail-safe operator
Specifies a fail safe operator.

Operator
Sets the name of the operator to receive fail-safe notifications.

Notify using
Sets the method to use for notifying the fail-safe operator.

Token Replacement
This section allows you to enable job step tokens that can be used in jobs run by SQL Server Agent alerts. For more information about job step tokens, see Use Tokens in Job Steps.

> **ⓘ Important**
>
> Any Windows user with write permissions on the Windows Event Log may be able to access job steps that are activated by SQL Server Agent alerts. To avoid this security risk, SQL Server Agent tokens that can be used in jobs activated by alerts are disabled by default. These tokens are: **$(A-DBN)**, **$(A-SVR)**, **$(A-ERR)**, **$(A-SEV)**, and **$(A-MSG)**.
>
> If you need to use these tokens, ensure that only members of trusted Windows security groups, such as the Administrators group, have write permissions on the Event Log of the computer where SQL Server resides before enabling them.

Replace tokens for all job responses to alerts
Select this check box to enable token replacement for jobs that are activated by SQL Server alerts.

See Also

Operators
Configure SQL Server Agent Mail to Use Database Mail
Database Mail

Feedback

Was this page helpful? Yes No

SQL Server Agent Properties (Connection Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view and modify the settings for the connection from the Microsoft SQL Server Agent service to SQL Server.

Options

Alias local host server
Specifies the alias to use to connect to the local instance of SQL Server. If you cannot use the default connection options for SQL Server Agent, define an alias for the instance and specify the alias here.

Use Windows Authentication
Sets Microsoft Windows Authentication as the authentication method used to connect to the SQL Server instance. SQL Server Agent connects as the account that the SQL Server Agent service runs as.

Use SQL Server Authentication
Sets SQL Server Authentication as the authentication method used to connect to the SQL Server instance.

> **Important**
>
> SQL Server Authentication is provided for backward compatibility. For improved security, use Windows Authentication if possible.

Login
Specifies the login name to use to connect to SQL Server.

Password

Specifies the password to use to connect to SQL Server.

Feedback

Was this page helpful? Yes No

SQL Server Agent Properties (General Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view and modify the general properties of the Microsoft SQL Server Agent service.

Options

Service state
Displays the current state of the SQL Server Agent service.

Auto restart SQL Server if it stops unexpectedly
SQL Server Agent restarts SQL Server if SQL Server stops unexpectedly.

Auto restart SQL Server Agent if it stops unexpectedly
SQL Server restarts SQL Server Agent if SQL Server Agent stops unexpectedly.

Filename
Specify the file name for the error log.

...
Browse to the error log file.

Include execution trace messages
Includes execution trace messages in the error log. Trace messages provide detailed information on SQL Server Agent operation. Therefore, the log file requires more disk space when this option is selected. This option should only be selected while troubleshooting a problem that may involve SQL Server Agent.

Write OEM file
Writes the error log file as a non-Unicode file. This reduces the amount of disk space

consumed by the log file. However, messages that include Unicode data may be more difficult to read when this option is enabled.

Net send recipient
Type the name of an operator to receive net send notification of messages that SQL Server Agent writes to the log file.

See Also

Operators
SQL Server Agent Error Log

Feedback

Was this page helpful? Yes No

SQL Server Agent Properties (History Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view and modify settings for managing the Microsoft SQL Server Agent service history log.

Options

Limit size of job history log
Sets limits for the amount of job history information that SQL Server Agent retains in the log.

Maximum job history log size (in rows)
Sets the maximum number of rows that SQL Server Agent retains. When the log grows to contain this number of rows, SQL Server Agent removes the oldest rows in the log as new rows are entered.

Maximum job history rows per job
Sets the maximum number of rows that SQL Server Agent retains per job. When the history for a particular job grows to contain this number of rows, SQL Server Agent removes the oldest rows in the log as new rows are entered.

Remove agent history
Specifies that SQL Server Agent will remove entries that have been in the log longer than a specified length of time. This is a one-time execution to remove the history. If a reoccurring job is needed, create and schedule a maintenance plan with a cleanup job.

Older than
Sets the amount of time that SQL Server Agent will retain entries.

See Also

SQL Server Agent Error Log

Feedback

Was this page helpful? Yes No

SQL Server Agent Properties (Job System Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view and modify how the Microsoft SQL Server Agent Service manages jobs.

Options

Shutdown time-out interval (in seconds)
Specifies the number of seconds that SQL Server Agent waits for jobs to complete before shutting down. If the job is still running after the interval specified, SQL Server Agent forcefully stops the job.

Use a non-administrator proxy account
Sets a non-administrator proxy account for SQL Server Agent. Microsoft SQL Server 2008 (10.0.x) and later versions support multiple proxies, therefore this option is only applicable when managing SQL Server Agent versions prior to SQL Server 2008 (10.0.x).

User name
Type the name of the user for the non-administrator proxy account. SQL Server supports multiple proxies, therefore this option is only applicable when managing SQL Server Agent versions prior to SQL Server 2008 (10.0.x).

Password
Type the password of the user for the non-administrator proxy account. SQL Server 2005 (9.x) and later versions support multiple proxies, therefore this option is only applicable when managing SQL Server Agent versions prior to SQL Server 2005 (9.x).

Domain
Type the domain of the user for the non-administrative proxy account. SQL Server

supports multiple proxies, therefore this option is only applicable when managing SQL Server Agent versions prior to SQL Server 2008 (10.0.x).

See Also

Implement Jobs

Feedback

Was this page helpful? Yes No

Target Servers (Download Instructions Tab)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to display and update the download instructions for one or more target servers.

Options

Target Server
Select the server to view or modify instructions for.

Job
Select the job to view or modify instructions for.

Target Server
View the name of the server the download instructions apply to.

Operation
View the operation that the download instruction will perform.

Object Name
View the name of the object that will be affected by the download instruction.

Date Posted
View the local date and time that the instruction was posted.

Date Downloaded
View the local date and time that the target server downloaded the instruction. If an error occurred during download, an error is indicated.

Instruction download status
View the most recent status for the selected download instruction.

Delete

Delete the selected download instruction.

Clear

Clear the status of the selected download instruction.

Related content

- Automated Administration Across an Enterprise

Feedback

Was this page helpful? Yes No

Target Servers (Target Server Status Tab)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view the status of the target servers for this master server.

Options

Target Server
View the name of the target server.

Local time
View the date and time of the target server in the local time zone for that server.

Last polled
View the local date and time that the target server last polled the master.

Unread instructions
View the number of instructions that the target server has not yet received.

Status
View the status of the target server.

Force Poll
Select this button to force the selected target servers to poll the master server.

Force Defection
Select this button to force the selected target servers to defect from the master server.

Post Instructions
Post instructions to the selected target servers.

Enable Auto Refresh
Select this option to automatically refresh the information shown.

Refresh every

Specify how often the information on this page is refreshed.

Related content

- Automated Administration Across an Enterprise

Feedback

Was this page helpful? Yes No

Alerts

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Events are generated by SQL Server and entered into the Microsoft Windows application log. SQL Server Agent reads the application log and compares events written there to alerts that you have defined. When SQL Server Agent finds a match, it fires an alert, which is an automated response to an event. In addition to monitoring SQL Server events, SQL Server Agent can also monitor performance conditions and Windows Management Instrumentation (WMI) events.

To define an alert, you specify:

- The name of the alert.
- The event or performance condition that triggers the alert.
- The action that SQL Server Agent takes in response to the event or performance condition.

Naming an Alert

Every alert must have a name. Alert names must be unique within the instance of SQL Server and can be no longer than **128** characters.

Selecting an Event Type

An alert responds to an event of a specific type. Alerts respond to the following event types:

- SQL Server events
- SQL Server performance conditions

- WMI events

The type of the event determines the parameters that you use to specify the precise event.

Specifying a SQL Server Event

You can specify an alert to occur in response to one or more events. Use the following parameters to specify the events that trigger an alert:

- **Error number**

  SQL Server Agent fires an alert when a specific error occurs. For example, you might specify error number 2571 to respond to unauthorized attempts to invoke Database Console Commands (DBCC).

- **Severity level**

  SQL Server Agent fires an alert when any error of the specific severity occurs. For example, you might specify a severity level of 15 to respond to syntax errors in Transact-SQL statements.

- **Database**

  SQL Server Agent fires an alert only when the event occurs in a particular database. This option applies in addition to the error number or severity level. For example, if an instance contains one database that is used for production and one database that is used for reporting, you can define an alert that responds to syntax errors in the production database only.

- **Event text**

  SQL Server Agent fires an alert when the specified event contains a particular text string in the event message. For example, you might define an alert that responds to messages that contain the name of a particular table or a particular constraint.

Selecting a Performance Condition

You can specify an alert to occur in response to a particular performance condition. In this case, you specify the performance counter to monitor, a threshold for the alert, and the behavior that the counter must show if the alert is to occur. To set a performance condition, you must define the following items on the SQL Server Agent **General** page of the **New Alert** or the **Alert Properties** dialog box:

- **Object**

  The object is the area of performance to be monitored.

- **Counter**

  A counter is an attribute of the area to be monitored.

- **Instance**

  The SQL Server instance defines the specific instance (if any) of the attribute to be monitored.

- **Alert if counter** and **Value**

  The threshold for the alert and the behavior that produces the alert. The threshold is a number. The behavior is one **of the following: falls below**, **becomes equal to**, or **rises above a number specified for Value**. The **Value** is a number that describes the performance condition counter. For example, to set an alert to occur for the performance object **SQLServer:Locks** when the **Lock Wait Time** exceeds 30 minutes, you would choose **rises above** and **specify 30 as the value**.

  As another example, you might specify that an alert occurs for the performance object **SQLServer:Transactions** when the free space in **tempdb** falls below 1000 KB. To set this, you would choose the counter **Free space in tempdb (KB)**, **falls below**,and a **Value** of **1000**.

  > ⓘ **Note**
  >
  > Performance data is sampled periodically, which can lead to a small delay (a few seconds) between the threshold being reached and the occurrence of the performance alert.

  > ⓘ **Note**
  >
  > An Event log variable that stores the server name is limited to 32 characters. Therefore if the combined size of your host name and instance name is greater than 32 characters, you may get the following error:

```
Warning,[466] Failed to copy server name LONGNAMESQLSERV\LONGINSTANCENAME
while generating performance counter alerts.
```

Selecting a WMI Event

You can specify that an alert occur in response to a particular WMI event. To select a WMI event, you must define the following on the SQL Server Agent **General** page of the **New Alert** or the **Alert Properties** dialog box:

- **Namespace**

  SQL Server Agent registers as a WMI client to the WMI namespace that is provided to query for events.

- **Query**

  SQL Server Agent uses the Windows Management Instrumentation Query Language (WQL) statement provided to identify the specific event.

Following are links to common tasks:

To create an alert based on a message number

- SQL Server Management Studio
- Transact-SQL

To create an alert based on severity levels

- SQL Server Management Studio
- Transact-SQL

To create an alert based on a WMI event

- SQL Server Management Studio
- Transact-SQL

To define the response to an alert

- SQL Server Management Studio
- Transact-SQL

To create a user-defined event error message

- Transact-SQL

To modify a user-defined event error message

- Transact-SQL

To delete a user-defined event error message

- Transact-SQL

To disable or reactivate an alert

- SQL Server Management Studio
- Transact-SQL

See Also

sp_update_alert (Transact-SQL)

Feedback

Was this page helpful? Yes No

Alert Properties - New Alert (General Page)

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view and modify the general properties of Microsoft SQL Server Agent alerts.

Options

Name
Change the name of the alert.

Enable
Enable the alert. When the alert isn't enabled, the actions specified in the alert don't occur.

Type
Select the type of alert:

- **SQL Server event alert** responds to messages in the Microsoft Windows event log.
- **SQL Server performance condition alert** responds to a specific condition in a performance counter.
- **WMI event alert** responds to a Windows Management Instrumentation (WMI) event.

SQL Server Event Alert Options

Database name
Specify a database for the event, or **all databases** to respond to a message regardless of the database where the event occurs.

Error number
Specify that this event responds to an error, and specify the error number.

Severity
Specify that this event responds to any message with a specific severity level, and specify the severity level.

Raise alert when message contains
Filter events by a specific string. When this option is selected, the alert only responds to events that contain a specific string.

Message text
Specify the string to use to filter events.

SQL Server Performance Condition Alerts

Object
Specify the performance object to monitor.

Counter
Specify the counter within the performance object to monitor.

Instance
Specify the instance of the counter to monitor.

Alert if counter
Specify the behavior of the counter that the alert responds to. For example, you might want the alert to respond to a condition where the value of the **Free space in `tempdb` (KB)** counter falls below a certain value, or you might want the alert to respond to a condition where the **SQL Compilations/sec** rises above a certain value.

Value
Specify a value for the counter.

WMI Event Alert Options

Namespace
Specify the namespace to use for the WMI Query Language (WQL) statement. Only namespaces on the computer that runs SQL Server Agent are supported.

Query
Specify the WQL statement that identifies the event that the alert responds to.

Related content

- Alerts
- Using WQL with the WMI Provider for Server Events
- Create an Alert Using an Error Number
- Create an Alert Using Severity Level

Feedback

Was this page helpful? Yes No

Alert Properties - New Alert (Options Page)

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view and modify options for Microsoft SQL Server Agent alerts.

Options

E-mail
Include error text from the event, if any, in e-mail notifications.

Pager
Include error text from the event, if any, in pager notifications.

Net send
Include error text from the event, if any, in net send notifications.

Additional notification message to send
Type any additional text to include in notification messages.

Delay between responses
Specify a delay for repeated occurrences of the event. Some events might occur frequently during a short period of time. In this case, you might want to know that the event has occurred, but you might not want a response for every event. Use this option to specify a time-out. With a delay, after the alert responds to an event, SQL Server Agent waits for the delay specified before responding again, regardless of whether the event occurs during the delay.

Minutes
Specify a delay in minutes. To respond every time the event occurs, specify 0 minutes and 0 seconds.

Seconds

Specify a delay in seconds. To respond every time the event occurs, specify 0 minutes and 0 seconds.

Related content

- Alerts

Feedback

Was this page helpful? Yes No

Alert Properties - New Alert (Response Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to specify a job that you want to run and to obtain a list of operators to be notified in response to a Microsoft SQL Server Agent alert.

Options

Execute job
Enables the **Job list**, **New job** and **View job** options.

New job
Open the **New Job** dialog box. This button isn't available when **Execute job** isn't selected.

View job
View or modify the selected job. This option isn't available when **Execute job** isn't selected.

Notify operators
Enables the controls that allow you to add, remove, or change operators.

Operator list
Lists the operators to notify when an alert occurs. To specify a notification method, select the **E-mail**, **Pager**, or **Net send** check box that is displayed after the operator name.This option not available when **Notify operators** isn't selected.

E-mail
Use e-mail to notify the operator.

Pager
Use the pager e-mail address to notify the operator.

Net send

Use **net send** to notify the operator.

New operator

Displays the **New Operator** dialog box, which you can use to create a new operator.

View operator

Displays the **Properties** dialog box for the currently selected operator. You can view and modified operator properties on the **Operator Properties** dialog box.

Related content

- Alerts
- Create an Alert Using Severity Level
- Edit an Alert
- Delete an Alert

Feedback

Was this page helpful? Yes No

Alert Properties (History Page)

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view and modify the history of Microsoft SQL Server Agent alerts.

Options

Date of last alert
Displays the date that the specified event last occurred, or **(Never occurred)** if the event hasn't occurred since the alert was created.

Date of last response
Displays the date that the alert last responded to the event, or **(Never responded)** if the event hasn't occurred since the alert was created.

Number of occurrences
Total number of occurrences of the event since the alert was created, or the last time that the count was reset.

Reset count
Reset the information on this page.

Related content

- Alerts

Feedback

Was this page helpful? Yes No

Job Categories - Manage Job Categories

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use the **Job Categories** dialog box to add or delete job categories. Built-in job categories cannot be deleted.

Options

Name
The name of the job category.

Number of jobs in category
The number of jobs defined for this category.

View Jobs
Opens the **Properties** dialog box for the selected category, to list all jobs currently defined for that category.

Add
Opens the **New Job Category** dialog box, to add a new job category

Delete
Removes the selected job category. Only enabled for user-defined job categories.

Refresh
Queries the server for the current information.

To access the Job Categories dialog box

1. In **Object Explorer**, expand **SQL Server Agent**, right-click **Jobs**, and then click **Manage Job Categories**.

Feedback

Was this page helpful? Yes No

Job Categories Properties - New Job Category

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Lists the jobs in one job category or all job categories, and enables you to add a new job category.

Options

Name
Type the name of the new job category. If this is the job category Properties dialog, the category you are viewing is displayed here.

Jobs in this category
Displays all existing jobs in this category.

Show all jobs
List all jobs.

Select
Change the category of the job by checking or clearing it.

Job
Displays the name of the job.

See Also

Create Jobs

Feedback

Was this page helpful? Yes No

Jobs That Reference a Schedule

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This dialog allows you to view information about the jobs that reference a particular schedule.

Options

Schedule
Displays the name of the schedule you are viewing.

Selected
Read-only.

Name
Name of a job that uses this schedule.

Enabled
Read-only. Indicates whether this job is currently enabled.

Category
Job category.

See Also

Create and Attach Schedules to Jobs

Feedback

Was this page helpful? Yes No

Job Properties - New Job (Alerts Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view and organize the alerts for a Microsoft SQL Server Agent job.

Options

Alert list
Lists the alerts for this job.

Add
Create a new alert for the job.

Edit
Modify the selected alert definition.

Remove
Remove the selected alert from the job, and delete the alert.

See Also

Alerts
Implement Jobs

Feedback

Was this page helpful? Yes No

Job Properties - New Job (General Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view and modify the general properties of a Microsoft SQL Server Agent job.

Options

Name
Change the name of the job.

Owner
Select the owner for the job.

Category
Select the job category for the job.

...
View the jobs in the selected category.

Description
Change the description of the job.

Enabled
Enable the job. When the job is not enabled, the job does not run in response to a schedule or an alert, although you can still start the job using the **sp_start_job** stored procedure.

Source
Displays the master server for the job. Only available on the **Job PropertiesGeneral** page.

Created
Displays the date and time that the job was created. Only available on the **Job**

PropertiesGeneral page.

Last modified
Displays the date and time that the job was last modified. Only available on the **Job PropertiesGeneral** page.

Last executed
Displays the date and time that the job last started running. Only available on the **Job PropertiesGeneral** page.

View Job History
View the job history for the job. Only available on the **Job PropertiesGeneral** page.

See Also

Implement Jobs
Job Categories - Manage Job Categories

Feedback

Was this page helpful? Yes No

Job Properties - New Job (Notifications Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to set actions for Microsoft SQL Server Agent to perform when the job completes.

Options

E-mail
Select this option to send e-mail when the job completes. After selecting this option, choose the operator to notify and the condition that will trigger the notification: **When the job succeeds**; **When the job fails**; or **When the job completes**.

Page
Select this option to send e-mail to an operator's pager when the job completes. After selecting this option, specify the operator to notify and the condition that will trigger the notification: **When the job succeeds**; **When the job fails**; or **When the job completes**.

Net send
Select this option to use net send to notify an operator when the job completes. After selecting this option, specify the operator to notify and the condition that will trigger the notification: **When the job succeeds**; **When the job fails**; or **When the job completes**.

Write to the Windows Application event log
Select this option to write an entry in the application event log when the job completes. After selecting this option, specify the condition that will cause the entry to be written: **When the job succeeds**; **When the job fails**; or **When the job completes**.

Automatically delete job
Select this option to delete the job when the job completes. After selecting this option, specify the condition that will trigger deletion of the job: **When the job succeeds**; **When the job fails**; or **When the job completes**.

See Also

Implement Jobs
How to: Configure SQL Server Agent Mail to Use Database Mail

Feedback

Was this page helpful? Yes No

Job Properties - New Job (Schedules Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view and organize schedules for a Microsoft SQL Server Agent job.

Options

Schedule list
Lists the schedules for this job.

New
Create a new schedule. After you create the schedule, the schedule is added to the job.

Pick
Select a schedule from the existing schedules. Because a job and schedule must have the same owner, this option will only allow you to pick from schedules that you own.

Edit
Edit the selected schedule to change job schedule properties.

Remove
Remove the selected schedule from the job. If no other jobs use the schedule, the schedule is deleted from the database.

See Also

Implement Jobs

Feedback

Was this page helpful? Yes No

Job Properties - New Job (Steps Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view and organize job steps for a Microsoft SQL Server Agent job.

Options

Job step list
Lists the job steps for this job.

Move step
Moves the job a step up or down in the list.

Start step
Select the step that SQL Server Agent starts with when the job begins.

New
Create a new job step below the selected job step.

Insert
Create a new job step above the selected job step.

Edit
Change the selected job step.

Delete
Delete the selected job step. When job steps are deleted their output log is automatically deleted.

See Also

Implement Jobs

Feedback

Was this page helpful? Yes No

Job Properties - New Job (Targets Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to manage the target servers for the job.

Options

Target local server
Run the job on the local server. If there are no target servers enlisted, this is the only option available.

Target multiple servers
Run the job on one or more target servers. After selecting this option, choose the servers to run the job on. Available target servers are listed. Click a target server to select it.

See Also

Implement Jobs

Feedback

Was this page helpful? Yes No

Job Step Properties - New Job Step (Advanced Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view and change the properties of a SQL Server Agent job step.

Options

On success action
Sets the action for SQL Server Agent to perform if the job step succeeds.

Retry attempts
Sets the number of times that SQL Server Agent attempts to retry a failed job step.

Retry interval (minutes)
Sets the amount of time for SQL Server Agent to wait between retry attempts.

On failure action
Sets the action for SQL Server Agent to perform if the job step fails.

Options for Transact-SQL Job Steps

Output file
Sets the file to use for output from the job step. This option is available only to members of the **sysadmin** fixed server role.

...
Browse to the file to use for output from the job step.

View
In SQL Server, this button is disabled for viewing output files. Instead, use Notepad to view Job Step output files.

Append output to existing file
Append output to the existing contents of the file. Otherwise, the previous file contents are overwritten each time the job step runs.

Log to table
Logs Job Step output to the **sysjobstepslogs** table in the **msdb** database.

View
After the job step has run at least once, select **View** to view its output in the table.

Append output to existing entry in table
Appends output to the existing contents of the table. Otherwise, the previous table contents are overwritten each time the job step runs.

Include step output in history
Select this option to include output from the job step in the job history.

Run as user
If you're a member of the **sysadmin** fixed server role, you can select another SQL login to run this job step.

Options for Operating System (CmdExec) Job Steps

Output file
Sets the file to use for output from the job step.

...
Browse to the file to use for output from the job step.

View
In SQL Server, this button is disabled for viewing output files. Instead, use Notepad to view Job Step output files.

Append output to existing file
Appends the Job Step output to the previous file contents each time it runs.

Log to table
Logs Job Step output to the **sysjobstepslogs** table in the **msdb** database.

View
After the job step has run at least once, select **View** to view its output in the table.

Append output to existing entry in table
Appends output to the existing contents of the table. Otherwise, the previous table contents are overwritten each time the job step runs.

Include step output in history
Select this option to include output from the job step in the job history.

Options for PowerShell Job Steps

Output file
Sets the file to use for output from the job step.

...
Browse to the file to use for output from the job step.

View
In SQL Server, this button is disabled for viewing output files. Instead, use Notepad to view Job Step output files.

Append output to existing file
Appends the Job Step output to the previous file contents each time it runs.

Log to table
Logs Job Step output to the **sysjobstepslogs** table in the **msdb** database.

View
After the job step has run at least once, select **View** to view its output in the table.

Append output to existing entry in table
Appends output to the existing contents of the table. Otherwise, the previous table contents are overwritten each time the job step runs.

Include step output in history
Select this option to include output from the job step in the job history.

Options for Replication Queue Reader Job Steps

Server
Sets the server to use for a replication queue reader job step.

Database
Sets the database to use for a replication queue reader job step.

Options for SQL Server Analysis Services Job Steps

Output file
Sets the file to use for output from the job step. This option is available only to members of the **sysadmin** fixed server role.

...
Browse to the file to use for output from the job step.

View
In SQL Server, this button is disabled for viewing output files. Instead, use Notepad to view Job Step output files.

Append output to existing file
Append output to the existing contents of the file. Otherwise, the previous file contents are overwritten each time the job step runs.

Log to table
Logs Job Step output to the **sysjobstepslogs** table in the **msdb** database.

View
After the job step has run at least once, select **View** to view its output in the table.

Append output to existing entry in table
Appends output to the existing contents of the table. Otherwise, the previous table contents are overwritten each time the job step runs.

Include step output in history
Select this option to include output from the job step in the job history.

See Also

- Manage Job Steps

Feedback

Was this page helpful? Yes No

Job Step Properties - New Job Step (General Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view and change the properties of a Microsoft SQL Server Agent job step, or to define a new job step.

To navigate to this page, in SQL Server Management Studio Object Explorer, expand SQL Server Agent, right-click **Jobs**, click **New Jobs**, select the **Steps** page, and click **New**. You can also navigate to this page by right-clicking a job in Object Explorer, clicking **Properties**, selecting the **Steps** page, and clicking **New**, **Insert**, or **Edit**.

Options

Step name
Set the name of the job step.

Type
Set the subsystem that the job step uses. Based on the subsystem you choose, the options displayed for defining the job step change.

Run as
Set the proxy account for the job step. Members of the sysadmin fixed server role may also specify the **SQL Agent Service Account**.

Database
Set the database that the job step runs in. This option is not available for all job step types.

Command
Set the command that the job step runs.

Options for Transact-SQL Job Steps

Open
Load the command from a file.

Select All
Select the text of the command.

Copy
Copy the selected text to the Clipboard.

Paste
Paste the contents of the Clipboard.

Parse
Check the syntax of the command.

Options for ActiveX Script Job Steps

> **Important**
>
> The ActiveX Scripting subsystem will be removed from SQL Server Agent in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature.

VBScript
Specify Microsoft Visual Basic Scripting Edition as the language for the job steps.

JScript
Specify JScript as the language for the job steps.

Other
Type the name of the language for job steps written in another scripting language.

Open
Load the command from a file.

Select All
Select the text of the command.

Copy
Copy the selected text.

Paste
Paste the contents of the Clipboard.

Options for Operating System (CmdExec) Job Steps

Process exit code of a successful command
Type the exit code that the command returns to indicate success.

Open
Load the command from a file.

Select All
Select the text of the command.

Copy
Copy the selected text.

Paste
Paste the contents of the Clipboard.

Options for PowerShell Job Steps

Open
Load the script from a file.

Select All
Select the text of the script.

Copy
Copy the selected text.

Paste
Paste the contents of the Clipboard.

Options for Replication Distributor Job Steps

Select All
Select the text of the command.

Copy
Copy the selected text.

Paste
Paste the contents of the Clipboard.

Options for Replication Merge Job Steps

Select All
Select the text of the command.

Copy
Copy the selected text.

Paste
Paste the contents of the Clipboard.

Options for Replication Queue Reader Job Steps

Database
The database to use for the job step.

Select All
Select the text of the command.

Copy
Copy the selected text.

Paste
Paste the contents of the Clipboard.

Options for Replication Snapshot Job Steps

Select All
Select the text of the command.

Copy
Copy the selected text.

Paste
Paste the contents of the Clipboard.

Options for Replication Transaction-Log Reader Job Steps

Select All
Select the text of the command.

Copy
Copy the selected text.

Paste
Paste the contents of the Clipboard.

Options for SQL Server Analysis Services Command Job Steps

Server
Select the server where the job step runs.

Open
Load the command from a file.

Select All
Select the text of the command.

Copy
Copy the selected text.

Paste
Paste the contents of the Clipboard.

Options for SQL Server Analysis Services Query Job Steps

Server
Select the server where the job step runs.

Database
The database to use for the job step.

Open
Load the command from a file.

Select All
Select the text of the command.

Copy
Copy the selected text.

Paste
Paste the contents of the Clipboard.

Options for Integration Services Package Execution Job Steps

General Tab

Specify where the Integration Services (SSIS) package is located and what authentication method to use. The following options are available when you select this tab.

Package source
Specify where the SSIS package is stored. Choose one of the following:

- **SQL Server**
- **File system**
- **SSIS Package Store**

Server
Type the server name where the SSIS package is stored. This option is only available when **SQL Server** or **SSIS Package Store** is specified for **Package Source**.

Use Windows Authentication
Logins to SQL Server use Microsoft Windows Authentication.

Use SQL Server Authentication
Logins to SQL Server use SQL Server Authentication. If this method of authentication is selected, enter the appropriate **User name** and **Password**.

> ⓘ **Important**
>
> SQL Server Authentication is provided for backward compatibility. For improved security, use Windows Authentication if possible.

Package
Type the location of the package.

> **ⓘ Important**
>
> For password-protected SSIS packages, click the **Configurations** tab to enter the password in the **Package Password** dialog box. Otherwise, the SQL Server Agent job that executes the password-protected package will fail.

Configurations Tab

Specify configuration options for the SSIS package. The following options are available when this tab is selected.

Configuration files
Lists the configuration files for the package.

Add
Add a configuration file for the package.

Remove
Remove a configuration file for the package.

Move Up
Move the selected configuration file up.

Move Down
Move the selected configuration file down.

Command Files Tab

Select command files for the package. Command files are processed in the order in which they appear in the list. The following options are available when you select this tab.

Command files
Lists the command files for the package.

Add
Add a command file.

Remove
Remove the selected command file.

Move Up
Move the selected command file up.

Move Down
Move the selected command file down.

Data Sources Tab

View the data sources specified in the package on this tab.

Connection Manager
View the name of the data source.

Description
View the description of the data source.

Connection String
View the connection string for the data source.

Execution Options Tab

View or change the execution options for the package on this tab.

Fail package on validation warnings
Select this option for package execution to fail if validation warnings occur.

Validate package without executing
Select this option for the job step to validate, but not execute, the package.

Maximum concurrent executables
Maximum number of executable files that can run at one time.

Enable package checkpoints
Select this option for the job step to use package checkpoints.

Checkpoint file
Type the name of the package checkpoint file.

...
Browse to find the package checkpoint file.

Override restart options
Select this option to specify restart options for this job step that are different from the restart options specified in the package.

Restart option
Select the action to take when the package restarts.

Logging Tab

View or change the log providers for the package on this tab.

Log Provider
Select the ClassID for the log provider.

Configuration String
Type the configuration string for the log provider.

Remove
Removes the log provider.

Set Values Tab

View or change property values for the package on this tab.

Property path
View or change the path for the property.

Value
View or change the value for the property.

Remove
Removes the property.

Verification Tab

Select the verification options for the job step on this tab.

Execute only signed packages
Run only packages that have been signed. When this option is selected, the job step fails if the package is unsigned.

Verify package build
Run only packages with a specific build number. When this option is selected, the job step fails if the package does not have the specified build number.

Build
Type the build number of the package.

Verify package ID
Run only packages with a specific ID. When this option is selected, the job step fails if the package does not have the specified ID.

Package ID
Type the package ID.

Verify version ID
Run only packages with a specific version ID. When this option is selected, the job step fails if the package does not have the specified version ID.

Version ID
Type the version ID.

Command Line Tab

Specify command-line options for the package. The following options are available when this tab is selected.

Restore the original options
Use the command-line options set in this dialog.

Edit the command line manually
Specify options in the command-line window.

Command line
Type the command line options to use for this package.

See Also

Manage Job Steps
SQL Server Agent Jobs for Packages
Administering Replication Agents

Feedback

Was this page helpful? Yes No

Operator Properties (History Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view the date, time, and type of the most recent notifications sent to the operator.

Options

By e-mail
Most recent notification attempts made by e-mail, or **(Never e-mailed)** if this operator has never been notified by e-mail.

By pager
Most recent notification attempts made by pager, or **(Never paged)** if this operator has never been notified by pager.

By net send
Most recent notification attempts made by net send, or **(Never notified by net send)** if this operator has never been notified by net send.

See Also

Operators

Feedback

Was this page helpful? Yes No

Operator Properties - New Operator (General Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view and modify the general properties of Microsoft SQL Server Agent operators.

Options

Name
Change the name of the operator.

Enabled
Enable the operator. When not enabled, no notifications are sent to the operator.

E-mail name
Specifies the e-mail address for the operator.

Net send address
Specify the address to use for **net send**.

Pager e-mail name
Specifies the e-mail address to use for the operator's pager.

Pager on duty schedule
Sets the times at which the pager is active.

Monday - Sunday
Select the days that the pager is active.

Workday begin
Select the time of day after which SQL Server Agent sends messages to the pager.

Workday end

Select the time of day after which SQL Server Agent no longer sends messages to the pager.

See Also

Operators

Feedback

Was this page helpful? Yes No

Operator Properties - New Operator (Notifications Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to set the alerts and jobs that notify the operator.

Options

Alerts
View the alerts in the instance.

Jobs
View the jobs in the instance.

Alert list
Lists the alerts in the instance.

Job list
Lists the jobs in the instance.

The following options are available in both the alert list and the job list:

E-mail
Notify this operator using e-mail.

Pager
Notify this operator by sending e-mail to the pager address.

Net send
Notify this operator using **net send**.

See Also

Feedback

Was this page helpful? Yes No

Proxy Account Properties (References Tab)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This read-only page lists the job steps that reference a Microsoft SQL Server Agent proxy account.

Options

Job name
Names of jobs that use this proxy account.

Job step
Names of job steps in the job that use this proxy account

Subsystem
Subsystem this proxy account uses.

See Also

Create a SQL Server Agent Proxy

Feedback

Was this page helpful? Yes No

Proxy Account Properties - New Proxy Account (General Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view or change the properties of a Microsoft SQL Server Agent proxy account.

Options

Proxy name
Type the name of the proxy.

Credential name
Type the name of the credential for the proxy.

> **Note**
>
> The credential name provided must be the name of an existing credential. For information on creating credentials, see How to: Create a Proxy

...
Launches the **Select Credential** dialog.

Description
Type the description for the proxy.

Active to the following subsystems
Select the subsystems that the proxy account has access to.

Reassign job steps to
Select the proxy to reassign job steps to. This list is enabled when you revoke access to a subsystem that the proxy previously had access to.

See Also

Create a SQL Server Agent Proxy

Feedback

Was this page helpful? Yes No

Proxy Account Properties - New Proxy Account (Principals Tab)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view or change the principals that can use a Microsoft SQL Server Agent proxy account in job steps.

Options

Proxy account principals
Lists the principals that can use this proxy account.

Add
Adds a principal to the list.

Remove
Removes the selected principal from the list.

See Also

Create a SQL Server Agent Proxy

Feedback

Was this page helpful? Yes No

Register Servers

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Registering a server in SQL Server Management Studio allows you to store the server connection information for future connections.There are three ways to register a server in SQL Server Management Studio.

1. Local instances of SQL Server are automatically registered during the first launch of Management Studio after its installation.
2. You can also initiate the automatic registration process at any time, to restore the registration of local server instances.
3. Lastly, you can register a server using the Registered Servers tool in SQL Server Management Studio.

Benefits of Registered Servers

With Registered Servers you can:

- Register servers to preserve the connection information.
- Determine if a registered server is running.
- Easily connect Object Explorer and Query Editor to a registered server.
- Edit or delete the registration information for a registered server.
- Create groups of servers.
- Provide user-friendly names for registered servers by providing a value in the **Registered server name** box that is different from the **Server name** list.
- Provide detailed descriptions for registered servers.
- Provide detailed descriptions of registered server groups.
- Export registered server groups.
- Import registered server groups.
- View the SQL Server log files for online or offline instances of SQL Server.

Related Tasks

Use the following topics to get started with registered servers:

Expand table

Description	Topic
Register local server instances	Register a Connected Server (SQL Server Management Studio)
Register a server	Create a New Registered Server (SQL Server Management Studio)
View registered servers	View Registered Servers in SQL Server Management Studio
Remove a registered server	Remove a Registered Server (SQL Server Management Studio)
Change a server's registration	Change a Server's Registration (SQL Server Management Studio)
Connect to a registered server	Connect to a Registered Server (SQL Server Management Studio)
Disconnect from a registered server	Disconnect from a Registered Server (SQL Server Management Studio)
Move a registered server or server group	Move a Registered Server or Registered Server Group (SQL Server Management Studio)
Change the name of a registered server or server group	Change the Name of a Registered Server or Registered Server Group (SQL Server Management Studio)
Create or edit a server group	Create or Edit a Server Group (SQL Server Management Studio)
Remove a server group	Remove a Server Group (SQL Server Management Studio)
Export registered server information	Export Registered Server Information (SQL Server Management Studio)
Import registered server information	Import Registered Server Information (SQL Server Management Studio)
Create a Central Management Server and Server Group	Create a Central Management Server and Server Group (SQL Server Management Studio)
Execute statements against multiple servers simultaneously	Execute Statements Against Multiple Servers Simultaneously (SQL Server Management Studio)

See Also

Remote Servers

Feedback

Was this page helpful? Yes No

Registered Servers F1 Help

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This section contains the F1 Help for the Registered Servers component in SQL Server Management Studio. It describes the various options.

To learn about Registered Servers and get links to what to do with them, go to the Register Servers topic.

Click to save the Registered Servers settings.

Reporting Services New or Edit Server Registration (General Tab)

Use this tab to specify options when registering an instance of Reporting Services.

To access this page, in Registered Servers, click **Reporting Services** on the **Registered Servers** toolbar, right-click any registered servers group such as **Reporting Services**, point to **New**, and then click **Server Registration**.

Options

Server type
When a server is registered from Registered Servers, the **Server type** box is read-only, and it matches the type of server displayed in the **Registered Servers** pane. To register a different type of server, click the server you want on the **Registered Servers** toolbar before starting to register a new server.

Server name
Specify the report server instance to connect to. In Management Studio, you can access a report server through its instance name. You can have one report server instance for each SQL Server instance that you install. If you are using the default instance, type the name of the SQL Server instance. If you are using a named instance, specify the named instance to connect to the report server in the format MSSQL$InstanceName.

Authentication
Authentication to a report server is made through Internet Information Services (IIS).

Select from one of the following authentication modes when connecting to Reporting Services:

Windows Authentication Mode (Windows Authentication)
Connect to the report server instance using your Microsoft Windows credentials.

Basic Authentication
Connect using **Basic Authentication** if your Reporting Services installation is configured to use Basic Authentication.

Forms Authentication
Connect using **Forms Authentication** if your Reporting Services installation is configured to use a custom authentication extension.

User Name
Enter the login name to use for the connection. This option is only available if you have selected **Basic** or **Forms Authentication**.

Password
Enter the password for the user name. This option is only editable if you have selected **Basic** or **Forms Authentication**.

Remember password
Store the password you have entered. This option is only available if you have clicked **Basic** or **Forms Authentication**.

> ⓘ **Note**
>
> If you have stored the password and want to stop storing it, clear this check box and then click **Save**.

Registered server name
The name you want to appear in Registered Servers. This name does not have to match the name in the **Server name** box.

Registered server description
Enter an optional description of the server.

Test
Click to test the connection to the server selected in **Server name**.

Analysis Services - Multidimensional Data New or Edit Server Registration (General Tab)

Use this tab to specify options when registering an instance of Analysis Services.

To access this page, in Registered Servers click **Analysis Services** on the Registered Servers toolbar, right-click any registered servers group, such as **Analysis Services**, point to **New**, and then click **Server Registration**.

Options

Server type
When a server is registered from Registered Servers, the **Server type** box is read-only, and it matches the type of server displayed in the Registered Servers pane. To register a different type of server, click the server you want on the **Registered Servers** toolbar before starting to register a new server.

Server name
Select the server instance to connect to. The server instance last connected to is displayed by default.

Authentication
Windows Authentication allows a user to connect using their Microsoft Windows credentials, as a Windows user or as a member of a Windows group.

User name
This option is not available in this release.

Password
This option is not available in this release.

Remember password
This option is not available in this release.

Registered server name
The name you want to appear in Registered Servers. This name does not have to match the **Server name** box.

Registered server description
Enter an optional description of the server.

Test
Click to test the connection to the server selected in **Server name**.

SSIS New or Edit Server Registration (General Tab)

Use this tab to specify options when registering Integration Services.

To access this page, in Registered Servers, click **Integration Services** on the **Registered Servers** toolbar, right-click any registered servers group, point to **New**, and then click **Server Registration**.

Options

Server type
When a server is registered in Registered Servers, the **Server type** box is read-only, and it matches the type of server displayed in Registered Servers. To register a different type of server, click **Database Engine**, **Analysis Server**, **Reporting Services**, **SQL Server Compact Edition**, or **Integration Services** on the **Registered Servers** toolbar before starting to register a new server.

Server name
Select the server to which to connect. The server last connected to is displayed by default.

Authentication
Windows Authentication mode allows a user to connect through a Microsoft Windows user account.

User name
This option is not available in this release.

Password
This option is not available in this release.

Remember password
This option is not available in this release.

Registered server name
The name you want to appear in **Registered Servers**. This name does not have to match the **Server name** box.

Registered server description
Enter an optional description of the server.

Test

Click to test the connection to the server selected in **Server name**.

Feedback

Was this page helpful? Yes No

Generate Scripts (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

SQL Server Management Studio provides two mechanisms for generating Transact-SQL scripts. You can create scripts for multiple objects by using the **Generate and Publish Scripts Wizard**. You can also generate a script for individual objects or multiple objects by using the **Script as** menu in **Object Explorer**.

For a detailed Tutorial on scripting various objects using SQL Server Management Studio (SSMS), see Tutorial: Scripting in SSMS.

Before You Begin

Choose the mechanism that best meets your requirements.

Generate and Publish Scripts Wizard

Use the **Generate and Publish Scripts Wizard** to create a Transact-SQL script for many objects. The wizard generates a script of all the objects in a database, or a subset of the objects that you select. The wizard has many options for your scripts, such as whether to include permissions, collation, constraints, and so on. For instructions on using the wizard, see Generate and Publish Scripts Wizard.

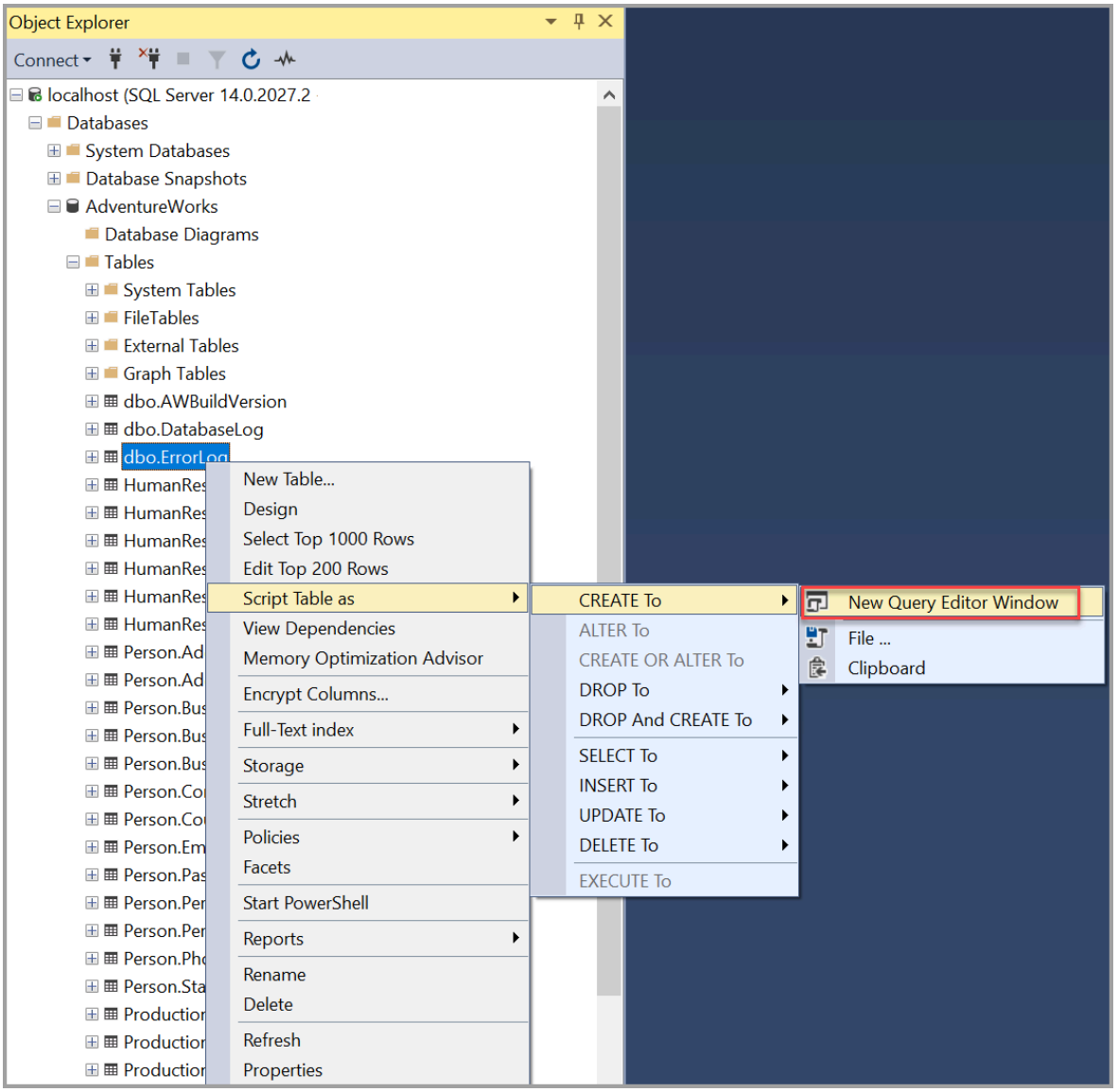
Object Explorer Script As Menu

You can use the **Object Explorer Script as** menu to script a single object, script multiple objects, or script multiple statements for a single object. You can choose one of several types of scripts; for example to create, alter, or drop the object. You can save the script in a Query Editor window, to a file, or to the Clipboard. The script is created in Unicode format.

To generate a script of a single object

To script a single object

1. In Object Explorer, connect to an instance of the SQL Server Database Engine and then expand that instance.
2. Expand **Databases**, and then expand the database containing the object to be scripted.
3. Expand the category of the object. For example, expand the **Tables** or **Views** node.
4. Right-click the object, point to **Script <object type> as**, For example, point to **Script Table as**.
5. Point to the script type, such as **Create to** or **Alter to**.
6. Select the location to save the script, such as **New Query Editor Window** or **Clipboard**.



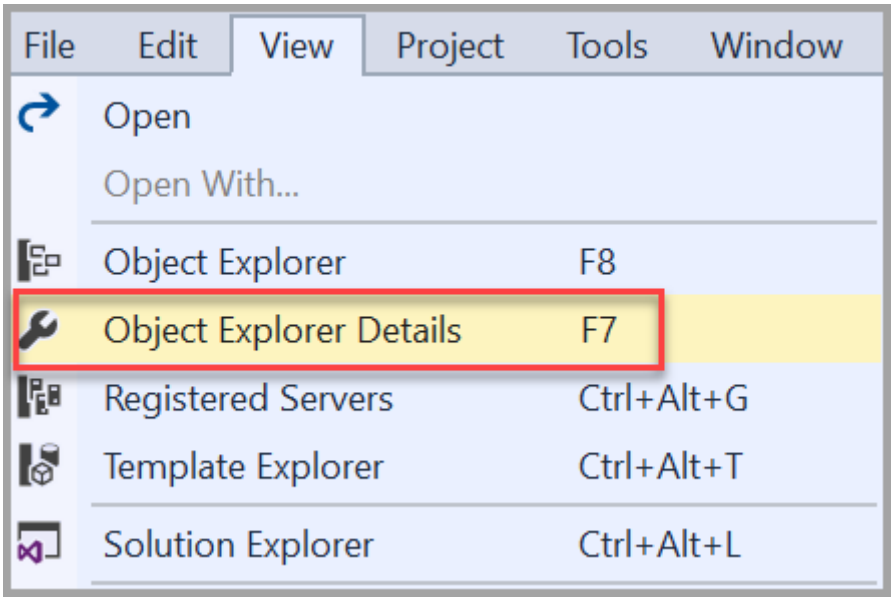
You can use the **Object Explorer Details** pane to generate a script for multiple objects of the same category.

1. In Object Explorer, connect to an instance of the SQL Server Database Engine and then expand that instance.

2. Expand **Databases**, and then expand the database containing the objects to be scripted.

3. Expand the category node of the types of object you want to script, such as the **Tables** node.

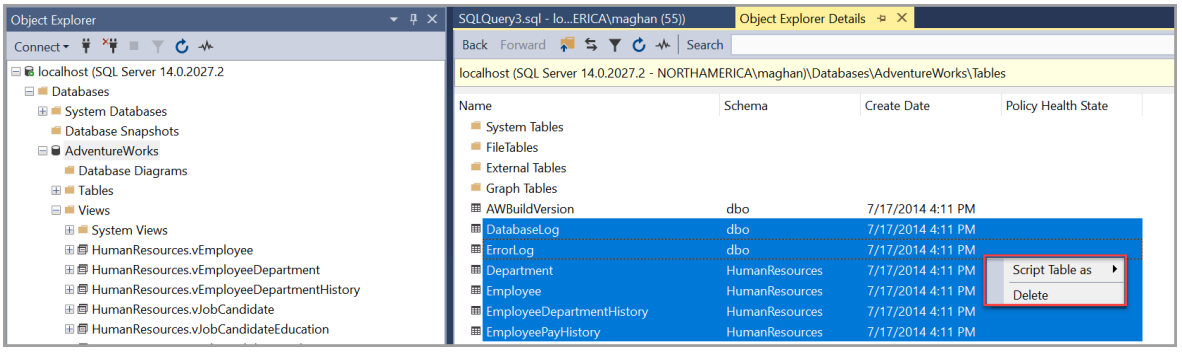
4. Open the **Object Explorer Details** pane by either selecting **F7**, or opening the **View** menu and selecting **Object Explorer Details**.



5. Left-click one of the objects you want to script.

6. Ctrl + left-click the second object you want to script.

7. Right-click one of the selected objects, and select **Script <object type> as**.



Feedback

Was this page helpful? Yes No

Generate Scripts Wizard

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This article teaches you to use the **Generate Scripts Wizard** to create scripts to transfer a database between instances of SQL Server, Azure SQL Database, or Azure SQL Managed Instance.

You can generate scripts for a SQL Server database in your local network, or from Azure SQL. The generated scripts can be run on another instance of SQL Server or on Azure SQL. You can create scripts for an entire database, or limit it to specific objects.

For a more detailed tutorial on using the Generate Scripts wizard, see Tutorial: Generate Scripts Wizard.

Prerequisites

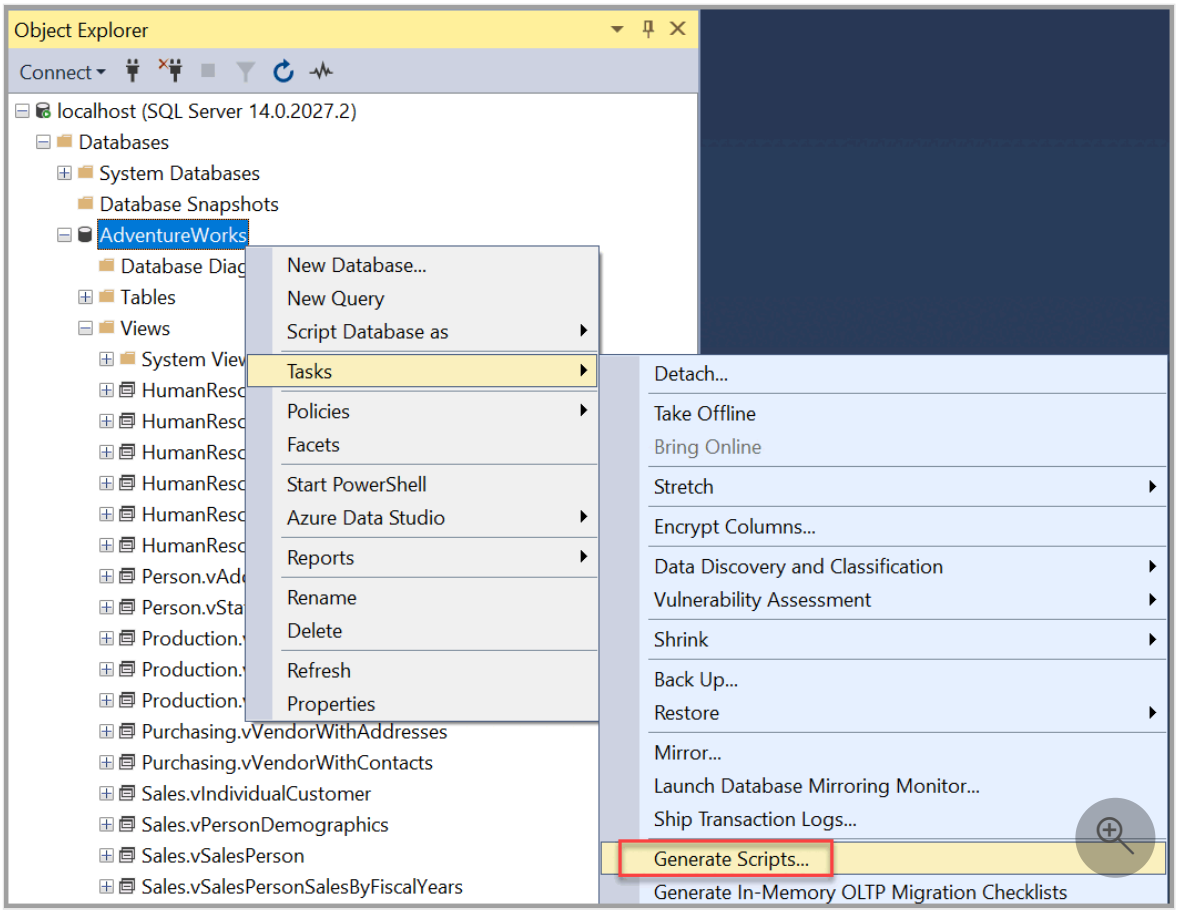
The source and target database can be on Azure SQL Database or Azure SQL Managed Instance, or an instance of the Database Engine running SQL Server 2005 (9.x) or later versions.

Permissions

The minimum permission to generate scripts is membership in the **db_ddladmin** fixed database role on the origin database.

Use the Generate Scripts Wizard

1. In **Object Explorer**, expand the node for the instance containing the database to be scripted.
2. Right-click on the database you want to script, and select **Tasks** > **Generate Scripts**.



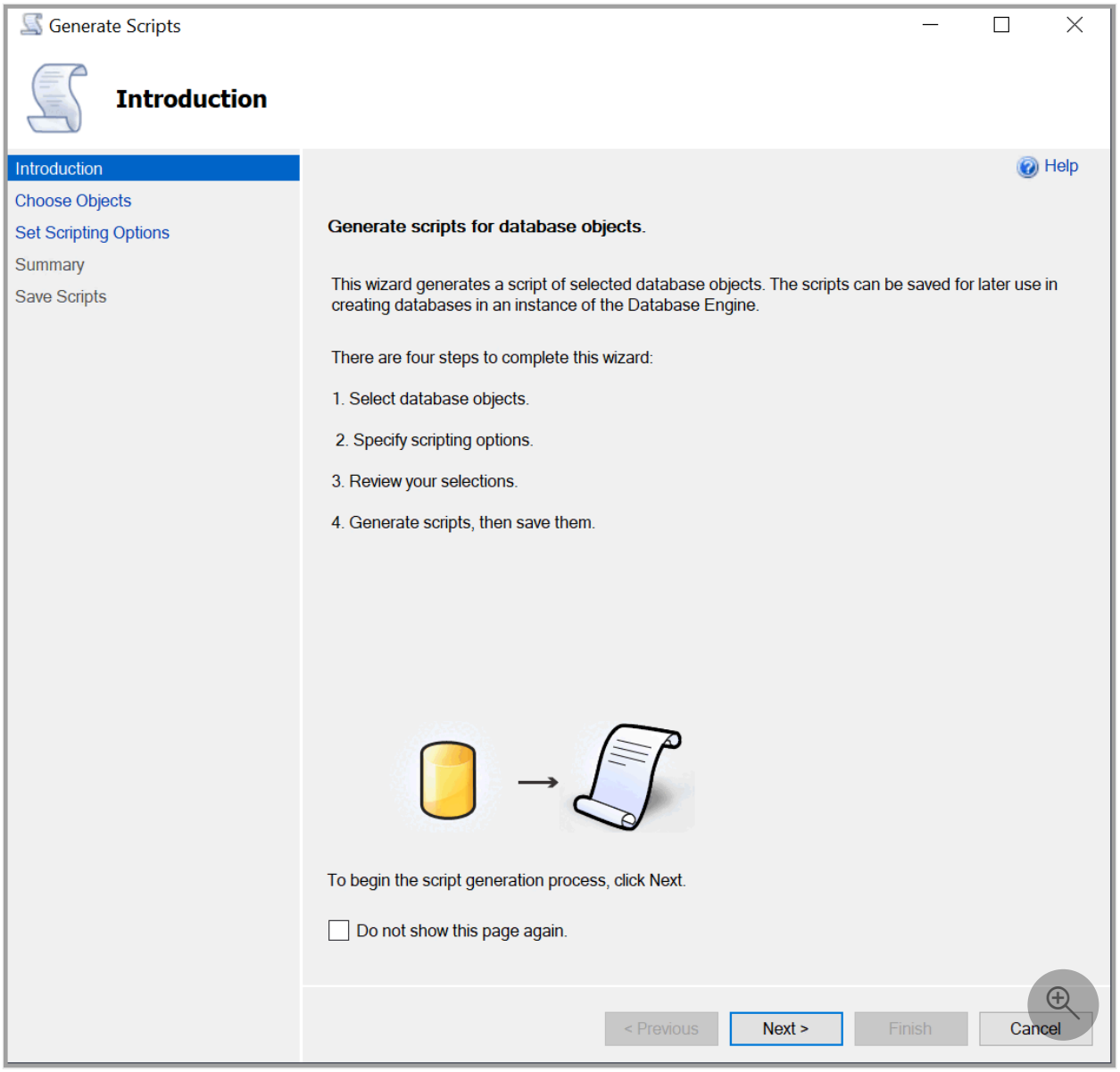
3. Complete the wizard dialogs:

   - Introduction page
   - Choose Objects page
   - Set Scripting Options page
   - Advanced Scripting Options page
   - Summary page
   - Save Scripts page

Introduction page

This page describes the steps for generating a script.

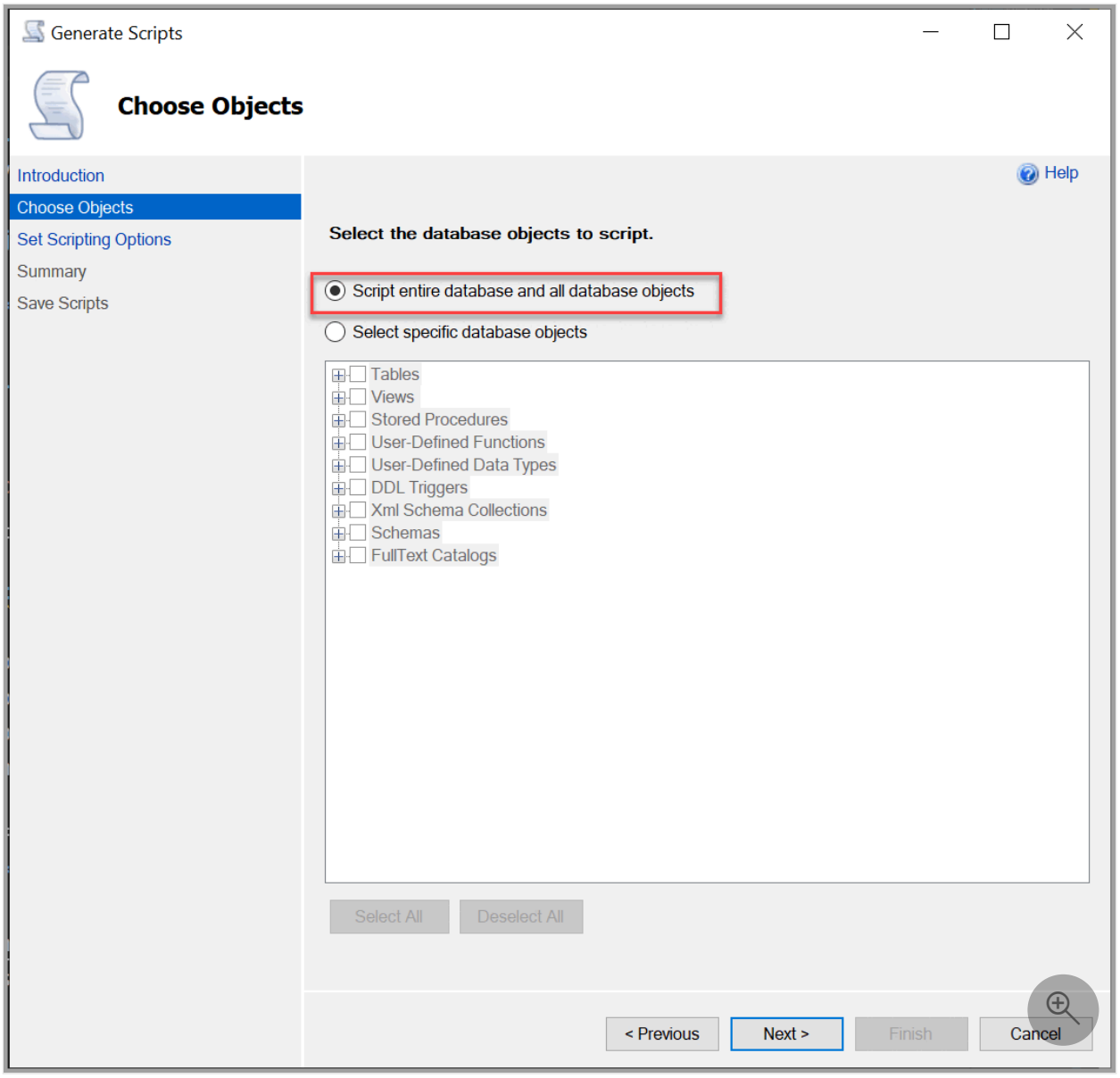
Do not show this page again - Skip this page the next time you start the **Generate Scripts Wizard**.



Choose Objects page

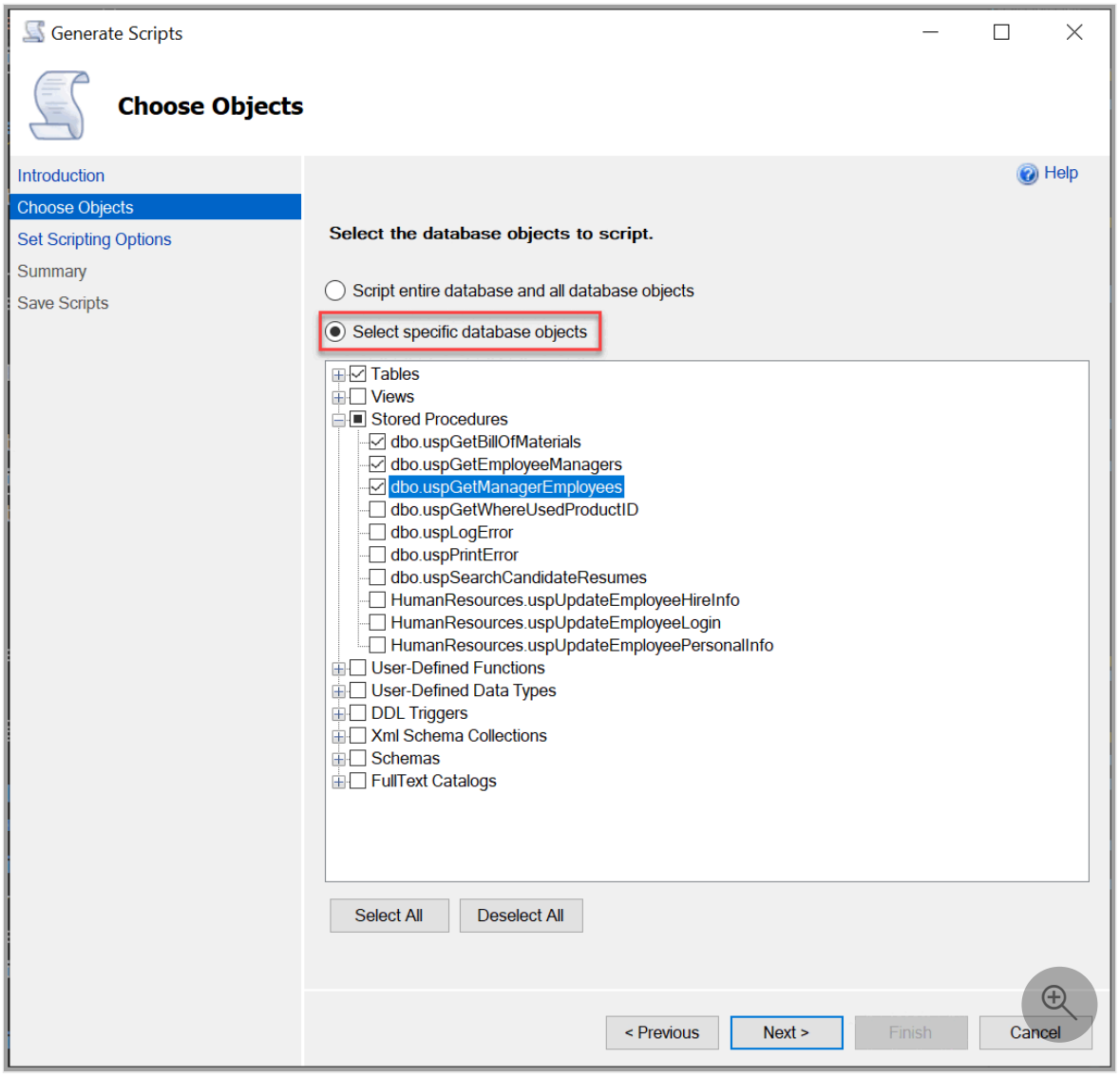
Use this page to choose which objects you want to include in the scripts generated by this wizard.

Script Entire Database Option - Select to generate scripts for all objects in the database and to include a script for the database itself.



Select specific database objects - Select to limit the wizard to generate scripts for only the specific objects in the database that you choose:

- **Database objects** - Select at least one object to include in the script.
- **Select All** - Selects all available check boxes.
- **Deselect All** - Clears all the check boxes. You must select at least one database object to continue.

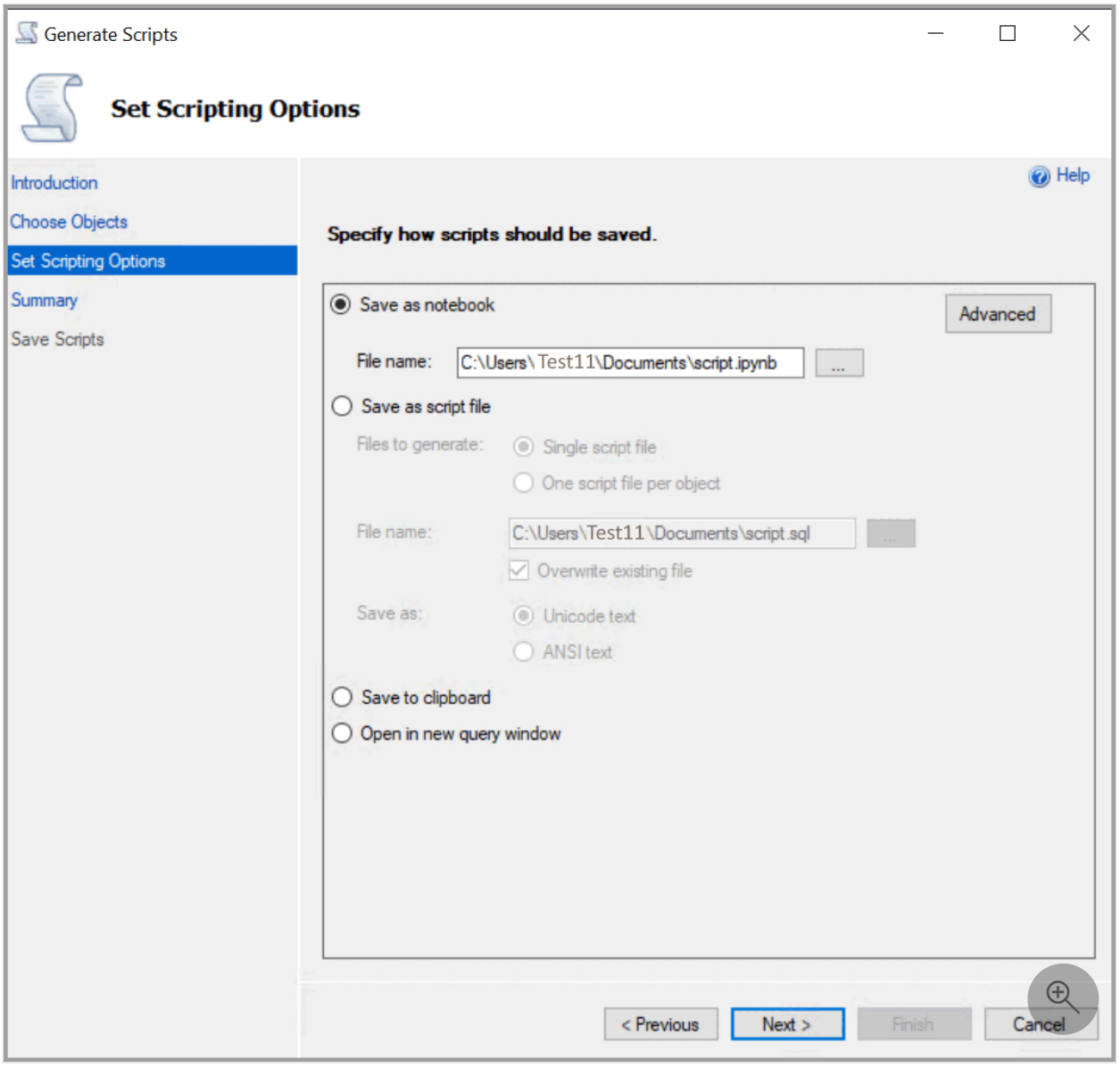


Set Scripting Options page

Use this page to specify if you want the wizard to save scripts to the location of your choice.

Options - If you want the wizard to save scripts to a location of your choice, select **Save scripts to a specific location**. You can later run the scripts against either an instance of the Database Engine, or against SQL Database.

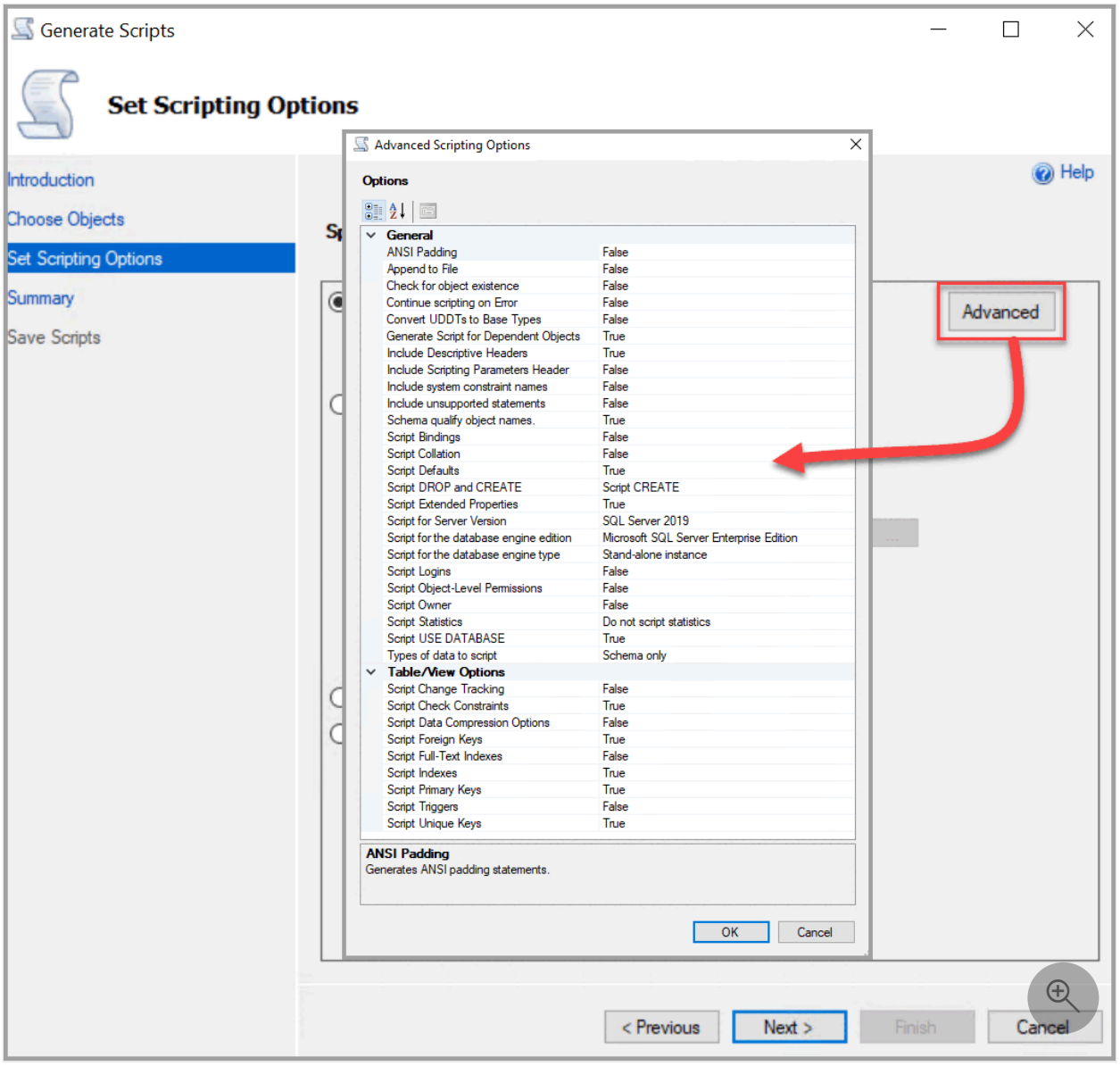
Save Scripts to a Specific Location - save one or more Transact-SQL script files to a location you specify.



- Save as notebook - Save the script to one or more .sql files. Select the browse button (...) to specify a name and location for the file.
- **Save as script file** Save the script to one or more .sql files. Select the browse button **(...)** to specify a name and location for the file. Select the **Overwrite existing file** check box to replace the file if one already exists with the same name. Select **Single script file** or **One script file per object** to specify how the scripts should be generated. Select **Unicode text** or **ANSI text** to specify the kind of text that should be used in the script.
- **Save to Clipboard** - Save the Transact-SQL script to the Clipboard.
- **Open in new query window** - Generate the script to a Database Engine Query Editor window. If no editor window is open, a new editor window is opened as the target for the script.
- **Advanced** - Display the **Advanced Options** dialog box where you can select advanced scripting options.

Advanced Scripting Options page

Use this page to specify how you want this wizard to generate scripts. Many different options are available. Options are greyed out if they aren't supported by the version of SQL Server or SQL Database specified in **Database engine type**.



Options - Specify advanced options by selecting a value from the list of available settings to the right of each option.

General - The following options apply to the entire script.

- **ANSI Padding** - Includes `ANSI PADDING ON` in the script. The default is **False**.
- **Append to file** - When **True**, this script is added to the bottom of an existing script, specified on the **Set Scripting Options** page. When **False**, the new script overwrites a previous script. The default is **False**.

- **Check for object existence** - When **True**, adds the existence check before generating the create statement for your SQL Objects. For example: tables, views, functions, or stored procedures. The `CREATE` statement is wrapped in an IF statement. If you know your target is clean, the script is a lot cleaner. If you do NOT expect the objects to exist on the target, you get an error. The default is **False**.

- **Continue scripting on error** - When **False**, scripting stops when an error occurs. When **True**, scripting continues. The default is **False**.

- **Convert UDDTs to base types** - When **True**, user-defined data types (UDDT) are converted into the underlying base data types that were used to create them. Use **True** when the UDDT doesn't exist in the database where the script runs. When **False**, UDDTs are used. The default is **False**.

- **Generate script for dependent objects** - Generates a script for any object that is required to be present when the script for the selected object is executed. The default is **True** for the **Script entire database** option, and **False** for **Select specific database objects**.

- **Include descriptive headers** - When **True**, descriptive comments are added to the script separating the script into sections for each object. The default is **False**.

- **Include if NOT EXISTS** - When **True**, the script includes a statement to check whether the object already exists in the database, and doesn't try to create a new object if the object already exists. The default is **False**.

- **Include system constraint names** - When **False**, the default value of constraints that were automatically named on the origin database are automatically renamed on the target database. When **True**, constraints have the same name on the origin and target databases.

- **Include unsupported statements** - When **False**, the script doesn't contain statements for objects that aren't supported on the selected server version or engine type. When **True**, the script contains the unsupported objects. Each statement for an unsupported object has a comment that the statement must be edited before the script can be run against the selected SQL Server version or engine type. The default is **False**.

- **Schema qualify object names** - Includes the schema name in the name of objects that are created. The default is **True**.

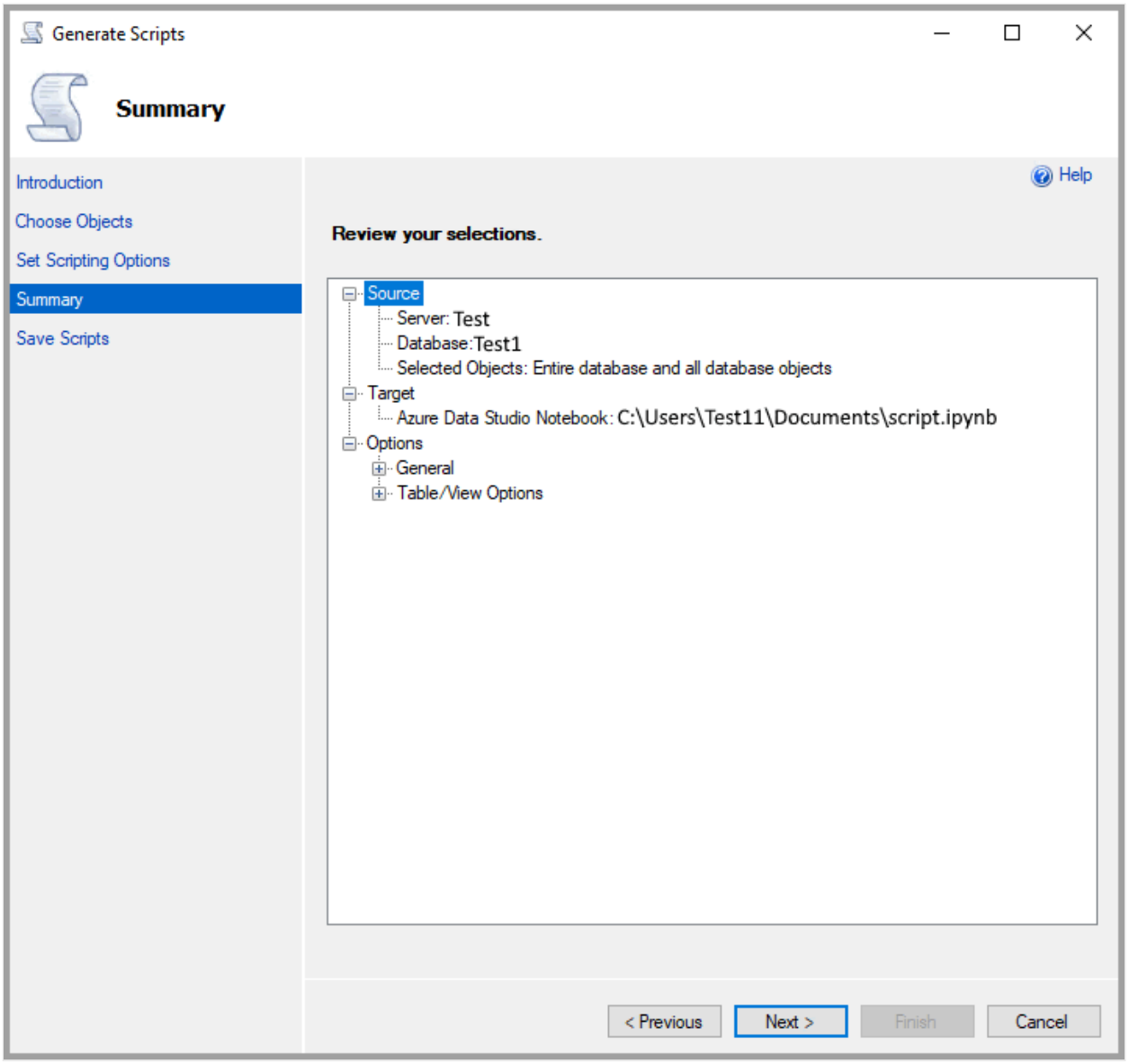
- **Script binding** - Generates a script for binding default and rule objects. The default is **False**. For more information, see CREATE DEFAULT (Transact-SQL) and CREATE RULE (Transact-SQL).

- **Script collation** - Includes collation information in the script. The default is **False**. For more information, see Collation and Unicode support.
- **Script defaults** - Includes default objects used to set default values in table columns. The default is **True**. For more information, see Specify default values for columns.
- **Script drop and create** - When **Script CREATE**, Transact-SQL statements are included to create objects. When **Script DROP**, Transact-SQL statements are included to drop objects. When **Script DROP and CREATE**, the Transact-SQL drop statement is included in the script, followed by the create statement, for each scripted object. The default is **Script CREATE**.
- **Script extended properties** - Includes extended properties in the script if the object has extended properties. The default is **True**.
- **Script for engine type** - Creates a script that can be run on the selected type of either SQL Database or an instance of the SQL Server Database Engine. Objects not supported on the specified type aren't included in the script. The default is the type of the origin server.
- **Script for server version** - Creates a script that can be run on the selected version of SQL Server. Features new to a version can't be scripted for earlier versions. The default is the version of the origin server.
- **Script logins** - When the object to be scripted is a database user, this option creates the logins on which the user depends. The default is **False**.
- **Script object-Level permissions** - Includes scripts to set permission on the objects in the database. The default is **False**.
- **Script statistics** - When set to **Script Statistics**, this option includes the `CREATE STATISTICS` statement to re-create statistics on the object. The **Script statistics and histograms** option also creates histogram information. The default is **Do not script statistics**. For more information, see CREATE STATISTICS (Transact-SQL).
- **Script USE DATABASE** - Adds the `USE DATABASE` statement to the script. To make sure that database objects are created in the correct database, include the `USE DATABASE` statement. If you expect to use the script in a different database, select **False** to omit the `USE DATABASE` statement. The default is **True**. For more information, see USE (Transact-SQL).
- **Types of data to script** - Selects what should be scripted: **Data only**, **Schema only**, or both. The default is **Schema only**.

Table/View Options - The following options apply only to scripts for tables or views.

- **Script change tracking** - Scripts change tracking, if enabled on the origin database, or tables in the origin database. The default is **False**. For more information, see About Change Tracking (SQL Server).
- **Script check constraints** - Adds `CHECK` constraints to the script. The default is **True**. `CHECK` constraints require data that is entered into a table to meet some specified condition. For more information, see Unique constraints and check constraints.
- **Script data compression options** - Scripts data compression options if they're configured on the origin database, or tables in the origin database. For more information, see Data compression. The default is **False**.
- **Script foreign keys** - Adds foreign keys to the script. The default is **True**. Foreign keys indicate and enforce relationships between tables.
- **Script full-text indexes** - Scripts the creation of full-text indexes. The default is **False**.
- **Script indexes** - Scripts the creation of indexes. The default is **True** for the **Script entire database** option, and **False** for **Select specific database objects**. Indexes help you find data quickly.
- **Script primary keys** - Scripts the creation of primary keys on tables. The default is **True**. Primary keys uniquely identify each row of a table.
- **Script triggers** - Scripts the creation of DML triggers on tables. The default is **False**. A DML trigger is an action programmed to execute when a data manipulation language (DML) event occurs in the database server. For more information, see DML Triggers.
- **Script unique keys** - Scripts the creation of unique keys on tables. Unique keys prevent duplicate data from being entered. The default is **True**. For more information, see Unique constraints and check constraints.

Summary page



This page summarizes the options that you selected in this wizard. To change an option, select **Previous**. To begin generating scripts, select **Next**.

Review your selections - Displays the selections you made for each page of the wizard. Expand a node to see the selected options for the corresponding page.

Save Scripts page

Use this page to monitor the progress of the wizard as it occurs.

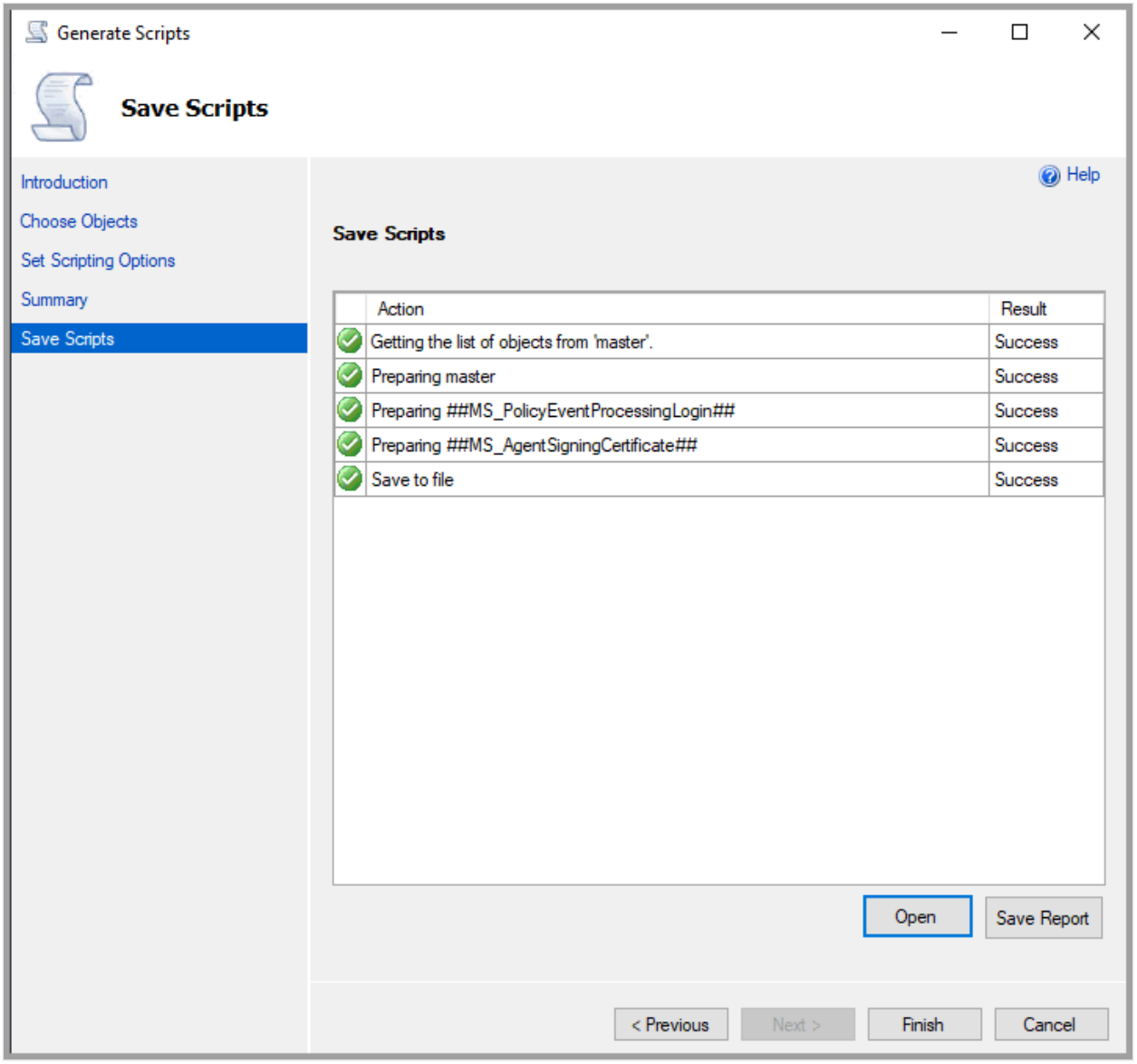
Details - View the **Action** column to see the progress of the wizard. After the wizard generates the scripts, it saves the scripts to a file. When each of these steps is complete, select the value in the **Result** column to see the outcome of the corresponding step.

Save Report - Select to save the results of the wizard's progress to a file.

Cancel - Select to close the wizard before processing is complete, or if an error occurs.

Finish - Select to close the wizard after processing is complete, or if an error occurs.

Save scripts



If all the settings are correct, your configuration finishes successfully.

Generate scripts on Azure Synapse Analytics

If the syntax generated when using "Script As..." doesn't look like Azure Synapse Analytics syntax or if you receive an error message, you might need to set your scripting options in SQL Server Management Studio to Azure Synapse Analytics.

How to set default scripting options to SQL Data Warehouse

In order to script objects with Azure Synapse Analytics syntax, set the default scripting option to Azure Synapse Analytics as follows:

1. Select **Tools** then **Options**.
2. Select **SQL Server Object Explorer**, and **Scripting**.
3. Under **Version options**, set **Match script settings to source** to **False**.
   a. Set **Script for database engine type** to **Microsoft Azure SQL Database Edition**.
   b. For dedicated SQL pools in Azure Synapse Analytics, set **Script for the database engine edition** to **Microsoft Azure SQL Data Warehouse Edition**.
4. Select **OK**.

How to generate scripts for SQL Data Warehouse when it's not the default scripting option

If you set Azure Synapse Analytics as your default scripting option as shown previously, these instructions can be ignored. However, if you choose to use different default scripting options, you might encounter an error. To avoid errors, follow these steps in order to Generate Scripts for Azure Synapse Analytics:

1. In **Object Explorer**, right-click your dedicated SQL pool under **Databases**.
2. Select **Generate Scripts**.
3. **Choose Objects** you wish to script.
4. In **Scripting Options**, select **Advanced**. Under **General** set:
   a. Script for the database engine type: **Microsoft Azure SQL Database**.
   b. Script for the database engine edition: **Microsoft Azure SQL Data Warehouse Edition**.
5. Select **Save Scripts** then **Finish**.

The options set in Step 4 aren't remembered. If you prefer for these options to be remembered, follow the instructions in How to set default scripting options to SQL Data Warehouse.

Related content

- Installing SMO
- Copy databases to other servers

Feedback

Was this page helpful? Yes No

Objects Supported by the Generate Scripts Wizard

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The Generate and Publish Scripts wizard supports a subset of the objects supported by the SQL Server Database Engine.

Supported Objects

The following table lists the objects that can be published supported by the Generate and Publish Scripts Wizard.

Application role

Database role

Schema

User-defined aggregate

View*

Assembly

DEFAULT constraint

Stored procedure*

User-defined data type

XML schema collection

CHECK constraint

Full-text catalog

Synonym

User-defined function

CLR (common language runtime) stored procedure*

Index

Table

User-defined table

CLR user-defined function

Rule

User**

User-defined type

*Published without encryption.

**Any non-system users that exist in the database are published as Roles.

Feedback

Was this page helpful? Yes No

XML Editor (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Provides a set of visual tools for working with XML Schemas, ADO.NET datasets, and XML documents. The XML Designer supports the XML Schema Definition (XSD) language defined by the World Wide Web Consortium (WC3). The designer does not support DTDs (document type definitions) or other XML schema languages, such as XDR (XML-Data Reduced).

To display the designer, add a dataset, XML Schema, or XML file to your project or open any of the file types listed in the table below.

> **Caution**
>
> There is no **Undo** command when working in Schema view. Plan your work carefully and save your files often.

The designer provides the following three views (or modes) to work on XML files, XML Schemas, and datasets:

Expand table

View	Description	File types supported
Schema	For visually creating and modifying XML Schemas and ADO.NET datasets.	.xsd
Data	For visually modifying XML data files in a structured data grid.	.xml
XML	For editing XML; the source editor provides color-coding and IntelliSense, including Complete Word and List Members.	.xml .xsd .xslt .wsdl.web.resx.tdl.wsf.hta.disco.vsdisco.config
ShowPlan	Displays xml query plans created using the SET SHOWPLAN_XML ON option.	.showplan

Schema View

Schema view provides a visual representation of the elements, attributes, types, and so on, that make up XML Schemas and ADO.NET datasets.

In Schema view you can construct schemas and datasets by dropping elements on the design surface from either the XML Schema tab of the Toolbox or from Server Explorer. Additionally, you can add elements to the designer by right-clicking the design surface and selecting Add from the shortcut menu.

In Schema view you can:

- Construct and modify existing XML Schemas and ADO.NET datasets
- Create and edit relationships between tables
- Create and edit keys
- Generate ADO.NET datasets from XML Schemas

> ⓘ **Note**
>
> The layout of elements in Schema view is stored in the .xsx file, which can be seen by clicking **Show All Files** in the Solution Explorer toolbar, and then expanding the .xsd file. If there is no .xsx file present, it means the .xsd file has never been opened in the XML Designer.

Customizing Schema View

The following features modify the visual layout of elements in Schema view:

- Zooming
- Expanding or collapsing of nested elements
- Automatically arranging layout of elements
- Resetting default state of collapsed elements

To expand hidden nested elements

- Click the plus icon on the bottom of the element.

To collapse nested elements

- Click the minus icon on the bottom-most element that you want to appear on the designer.

Data View

Data view provides a data grid that can be used to modify .xml files. Only the content (but not the tags and structure) in an XML file can be edited in Data view.

There are two separate areas in Data view: **Data Tables** and **Data**. The **Data Tables** area is a list of relations defined in the XML file, in the order of its nesting (from the outermost to the innermost). The **Data** area is a data grid that displays data based on the selection in the Data Tables area.

> ⓘ **Note**
>
> Newly created XML files contain no data and therefore cannot be displayed in Data view. There are also some instances of XML documents where data view cannot be invoked at all. Although the XML would be considered well formed, if it is not structured data trying to switch to Data view will generate the following message: "Although this document is well formed, it contains structure that Data View cannot display."

In Data view you can:

- Manually populate data tables
- Edit existing data tables
- Generate an XML Schema from an XML document

XML View

XML view provides an editor for editing raw XML and provides IntelliSense and color coding. Statement completion is available when working on .xsd files and .xml files that have an associated schema. Type < to initiate a tag and you will be presented with a list of elements that are valid at that location. After you type the element name and press the SPACEBAR, you will be presented with a list of attributes that the element supports.

> ⓘ **Note**

Microsoft IntelliSense options are not available on the toolbar. When in the XML Editor, to access the options, on the **Edit** menu, click **IntelliSense**.

SHOWPLAN view

Query plans can be saved in XML format when created using SET SHOWPLAN_XML ON option. Double-click a file with the .showplan extension to open the query plan.

See Also

Save an Execution Plan in XML Format

Feedback

Was this page helpful? Yes No

Color Coding in Query Editors

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The text entered in the code editors is assigned a category; each category is identified by a color. The colors help you quickly find text in your code. For example, comments stand out in dark green. The following table lists the most common colors. You can view the whole list of colors and their categories, and configure a custom color scheme by using the **Tools**, **Options** menu. For more information about how to change the default colors, see Change Font Color, Size, and Style.

Default Code Colors

Expand table

Color	Category
Red	SQL string
Dark green	Comment
Black on silver background	SQLCMD command
Magenta	System function
Green	System table, view, or table-valued function. Also, the system schemas sys and INFORMATION_SCHEMA.
Blue	Keyword
Teal	Line numbers or template parameter
Maroon	SQL Server stored procedure
Dark gray	Operators

Status Bar

You can configure registered servers or Database Engine servers in Object Explorer to have different colors in the Database Engine Query Editor status bar. This helps you identify which server each editor window is connected to when you have many windows

open at the same time. For more information about setting status bar colors, see Status Bar (Database Engine Query Editor).

Some types of editors do not display the status bar, or do not support multiple colors.

Feedback

Was this page helpful? Yes No

Code outlining

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can use the outlining feature in the SQL Server Management Studio query editors to selectively hide code when you edit queries. This enables you to more easily view the code you're working on, especially in large query files.

Outlining overview

By default, all code is visible when you open a query editor window. Regions of the code can be collapsed to hide it from view. A vertical line on the left edge of the editor window uses a square with a minus sign (`-`) to identify the start of each collapsible code region. When you select a minus sign, the text of the code region is replaced with a box that contains three periods (`...`), and the minus sign changes to a plus sign (`+`). When you select a plus sign, the collapsed code appears and the plus sign changes to a minus sign. When you move the pointer over a box that has three periods, a tooltip appears that shows the code in the collapsed section.

Enable or disable code outlining

You can manage this setting by navigating to **Tools** > **Options**, expanding the **Text Editor** section, and selecting **IntelliSense**. To disable code outlining, clear the **Outline statements** checkbox. Code outlining is enabled by default.

System outline regions

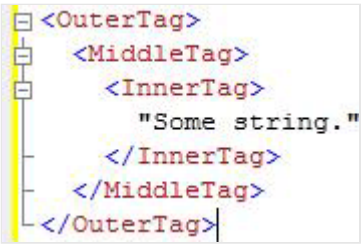
Each SQL Server Management Studio editor generates a set of default, system-defined outline regions.

The MDX and DMX code editors create outline regions for each multiline statement. This is the only level of outlining that these editors support.

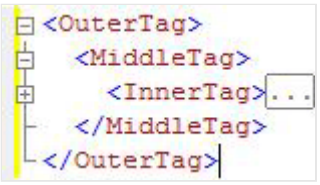
Analysis Services XMLA query editor regions

The Analysis Services XMLA Query Editor generates an outline region for each multiline XML attribute. The editor nests the outline regions for nested tags. For example, the

XMLA Editor creates three outline regions for the following document.

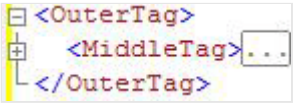


When you select the minus sign on the `<InnerTag>` line, just the `InnerTag` is collapsed, as shown in the following illustration.

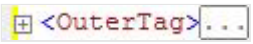


When you move the pointer over the box that has the three periods (`...`), the code in the collapsed region appears in a tooltip, as shown in the following illustration.

When you select the minus sign on the `<MiddleTag>` line, both the `MiddleTag` and `InnerTag` are collapsed, as shown in the following illustration.



When you select the minus sign on the `<OuterTag>` line, all three lines are collapsed, as shown in the following illustration.



Database Engine query editor regions

The Database Engine Query Editor generates outline regions for each element in the following hierarchy:

1. Batches. The first batch is the code from the start of the file to either the first `GO` command or the end of the file when there are no `GO` commands. After the first `GO`, there's one batch from each `GO` command to either the next `GO` command or the end of the file.

2. Blocks delimited by the following keywords:

   - `BEGIN` - `END`
   - `BEGIN TRY` - `END TRY`
   - `BEGIN CATCH` - `END CATCH`

3. Multiline statements.

For example, the Database Engine Query Editor creates three outline regions for the following query:

SQL

```
CREATE PROCEDURE Sales.SampleProc --Outline region 1
AS
BEGIN --Outline region 2
    SELECT GETDATE() AS TimeOfQuery;

    SELECT * --Outline region 3
    FROM sys.transmission_queue;

    SELECT @@VERSION;
END;
GO
```

You can select the minus sign on the `SELECT *` line to collapse just that `SELECT` statement. To collapse the whole `BEGIN - END` block, select the minus sign on the `BEGIN` line. To collapse the whole batch to the `GO` command, select the minus sign on the `CREATE PROCEDURE` line. You can't collapse the `SELECT GETDATE()` or `SELECT @@VERSION` lines individually because they are single line statements and don't get outlining regions.

Feedback

Was this page helpful? Yes No

Drag and Drop Text

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can select text and then drag and drop it in another location. You can drag and drop text:

- From Object Explorer into the editor to create a query.
- From one location to a different location in the current editor.
- Between editors.
- To the Microsoft Windows Recycle Bin.

To drag and drop text

1. Select the text you want to move, either with the mouse or with the keyboard.
2. Left-click the highlighted text and continue holding down the mouse button.
3. Move the mouse cursor to the destination where you want to place the text.
4. Release the mouse button to drop the text.

Dragging text moves it; that is, the text is erased from the old location and moved to the new location. Pressing the CTRL key while dragging text copies it to the new location. Dragging an object from Object Explorer is an exception and does not delete the object name from Object Explorer.

See Also

Navigate Code and Text

Feedback

Was this page helpful? Yes No

Search and Replace

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

There are several different ways to find and replace text. On the **Edit** menu, **Find and Replace** offers four choices: **Quick Find**, **Quick Replace**, **Find in Files**, or **Replace in Files**. Each of these opens versions of the **Find and Replace** dialog box. You can also search without a dialog box by using incremental search keyboard shortcut keys. These techniques allow you to control the scope of find and replace, and choose the method of reviewing search matches and replacements.

You should consider the following when you search and replace text:

- Options set in the **Find and Replace** dialog box affect all the searches. These options include **Match case**, **Match whole word**, **Search up**, **Search hidden text**, **Wildcards**, **Regular Expressions**, **Look in All Open Documents**, and **Look in Current Project**. All options are not available in all versions of the **Find and Replace** dialog box.
- **Undo** is available only for documents left open after a replace operation.
- **Undo** for a **Replace All** operation that spans more than one file is considered a single, bulk action across all the affected files. That is, you cannot undo the change in some files while retaining the change in other files.

Generally, you cannot search items with graphical views.

See Also

Search an Active Document Incrementally
Search Documents Interactively
Search Documents Using Results Lists
Search Text with Regular Expressions

Feedback

Was this page helpful? Yes No

Search an Active Document Incrementally

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can search a single document or window incrementally by entering text. The search operation highlights the first set of characters that matches the characters entered during the incremental search in the document or window. Incremental search automatically searches all of the text within a document or window except for text that has been hidden.

For the **Match case** option, incremental search uses the criteria from your previous search. For example, if you searched across multiple files using the **Find in Files** dialog box and select **Match Case**, and you next search incrementally, the search will be case-sensitive.

To search incrementally

1. Open the file or window to you want to search.
2. On the **Edit** menu, point to **Advanced**, and then click **Incremental Search**.

   The cursor icon changes to a binocular with an arrow, indicating the search direction, and the status bar displays "Incremental Search."
3. Begin typing the text string.

   The status bar displays the text you are entering while the editor highlights the first occurrence that matches the text. As you continue typing, the editor moves to the next match and highlights it. If no matches are available, the status bar displays the following.

   ```
   Incremental Search: <text> (not found)
   ```

Incremental searches are performed from the current location in the document downwards from left to right. Incremental searches can be done using keyboard shortcut keys.

> **Note**
>
> For a complete list of keyboard shortcut keys, see SQL Server Management Studio Keyboard Shortcuts.

See Also

Search and Replace
Search Documents Interactively
Search Documents Using Results Lists
Search Text with Regular Expressions

Feedback

Was this page helpful? Yes No

Search Documents Interactively

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Using the **Find and Replace** dialog box, you can search one or more open files or windows and move through the search matches one by one. This technique allows you to review each individual search match in the context of the text around the match. You also have the option of performing bulk find operations and reviewing search matches in report format using the **Find and Replace** dialog box.

To search all open documents

1. Open the items you wish to search.
2. On the **Edit** menu, point to **Find and Replace,** and then click **QuickFind**.
3. In the **Find and Replace** text box, enter the text to search for.
4. In the **Look in** list, select **All Open Documents**.

   > **Note**
   >
   > Certain open files might not be searched when **All Open Documents** is selected. Only files currently open in a textual view, such as Code view, are included in searches. Files in Designer view are not included in searches.

5. Select additional search options to increase the accuracy of the search.
6. Click **Find Next** to start the search, and continue clicking **Find Next** until the last file has been searched.

When the search passes the beginning or end of the document, a message is displayed in the status bar. A message box appears when the search reaches the starting point of the search.

To replace in all active files interactively

1. On the **Edit** menu, point to **Find and Replace**, and then click **QuickReplace**.
2. In the **Find what** box, enter the text to search for.

3. In the **Replace with** box, enter the text to replace the search text.
4. In the **Look in** list, select **All Open Documents**.
5. Click **Replace**, and continue clicking **Replace** until the last match in the last file has been replaced. Click **Find Next** to skip a match you do not want to replace.

   -or-

   Choose **Replace All** to replace all matches. A message box appears, listing the total number of replacements.

The **Replace All** command replaces all search matches, including those you have skipped with the **Find Next** button. To cancel **Replace All**, click **Undo** from the **Edit** menu before closing any of the files.

See Also

Search an Active Document Incrementally
Search and Replace
Search Documents Using Results Lists
Search Text with Regular Expressions

Feedback

Was this page helpful? Yes No

Search Documents Using Results Lists

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Using the **Find and Replace** dialog box, you can search and replace text in all files in a project or solution or in a file system folder, even when they are not open in SQL Server Management Studio. Matches from searches that were performed with the **Find and Replace** dialog box appear in the Find Results 1 and Find Results 2 windows, which allows you to view the exact text from the line containing the match.

To search in multiple files

1. On the **Edit** menu, point to **Find and Replace,** and then click **Find in Files**.
2. In the **Find what** text box, enter the text to search for.
3. In the **Look in** list, click **All Open Documents**, **Current Project**, **Entire Solution**, or type a directory path.
4. In the **File types** list, select one of the listed sets of file extensions or enter the extensions for the types of files to be searched, separated by semicolons. Use *.* to search all files in the directory listed in the **Look in** drop-down list.
5. Select from the remaining search options to improve the accuracy of the search.
6. Click **Find** to begin the search.

The matches for the search appear in the Find Results 1 window by default. You can browse the search matches by double-clicking each entry in the Find Results 1 window. To view the search results in a new window, select **Display in Find 2**.

To replace across multiple files or folders

1. On the **Edit** menu, point to **Find and Replace,** and then click **Replace in Files**.
2. In the **Find what** text box, enter the text to search for.
3. In the **Replace with** box, enter the text to replace the search text.
4. In the **Look in** list, click **All Open Documents**, **Current Project**, **Entire Solution**, or type a directory path.

5. Click **Replace** to replace the current search match with the text in the **Replace with** box. You can skip a single match by clicking **Find Next** or skip an entire file clicking **Skip File**.

   - or -

   Choose **Replace all** to replace all search matches with the text in the **Replace with** box. Select **Keep modified files open after Replace All** if you want to undo some of the replacements at another time.

   > ⓘ Note
   >
   > **Replace all** replaces all search matches, including those in files you have skipped with **Skip File** or **Find Next**. You can only use **Undo** for replacements made in files that remain open after the replacement operation.

Replacement information appears in the Find Results 1 window by default. You can browse replacements by double-clicking each entry in the Find Results 1 window.

See Also

Search and Replace
Search Documents Interactively
Search Text with Regular Expressions

Feedback

Was this page helpful? Yes No

How to search text with regular expressions

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The following regular expressions can replace characters or digits in the **Find what** field of the SQL Server Management Studio **Find and Replace** dialog box.

Prerequisites

- Download SSMS

How to enable regular expressions

Here are steps to enable regular expressions in search.

1. Go to **Edit** > **Find and Replace** > **Quick Find**.
2. Next to the search bar select the *down arrow* > **Find in Files**.
3. In the **Find and Replace** window expand **Find options** and select **Use Regular Expressions**.

The **Expression Builder** button next to the **Find what** field then becomes available. Select this button to display a list of the available regular expressions. When you choose any item from the **Expression Builder**, it's inserted into the **Find what** string.

The following table describes some of the regular expressions in the **Expression Builder**.

Expand table

Expression	Description
`.`	Match any single character (except a line break)
`.*`	Match any character zero or more times
`.+`	Match any character one or more times
`[abc]`	Match any character in the set `abc`
`[^abc]`	Match any character not in the set `abc`

Expression	Description
`\d`	Match any numeric character
`(?([^\r\n])\s)`	Match any whitespace character
`\b`	Match at the beginning or end of the word
`^`	Match at beginning of line
`.$`	Match any line break
`\w\r?\n`	Match a word character at end of line
`(dog | cat)`	Capture and implicitly number the expression `dog | cat`
`(?<pet>dog | cat)`	Capture subexpression `dog | cat` and name it `pet`

Examples

Some examples of using regular expressions.

Example 1: Finding All Select Statements

You want to find all SELECT statements in your SQL scripts.

SQL

```
SELECT\s+.*\s+FROM
```

Explanation for example 1

- SELECT\s+: Matches the word "SELECT" followed by one or more whitespace characters.
- .*: Matches any character (except for line terminators) zero or more times.
- \s+FROM: Matches one or more whitespace characters followed by the word "FROM".

Example 2: Finding Procedures with Specific Naming Patterns

You want to find all stored procedures that start with "usp_" in your SQL scripts.

```
```

```SQL
CREATE\s+PROCEDURE\s+usp_[A-Za-z0-9_]+
```

Explanation for example 2

- CREATE\s+PROCEDURE\s+: Matches the words "CREATE PROCEDURE" followed by one or more whitespace characters.
- usp_: Matches the literal string "usp_".
- [A-Za-z0-9_]+: Matches one or more alphanumeric characters or underscores.

Example 3: Finding Comments in SQL Scripts

You want to identify all single-line comments (starting with --) in your SQL scripts.

```SQL
--.*
```

Explanation for example 3

- --: Matches the literal string "--".
- .*: Matches any character (except for line terminators) zero or more times.

Example 4: Finding All Update Statements

You want to find all the UPDATE statements in your SQL scripts.

```SQL
UPDATE\s+.*\s+SET
```

Explanation for example 4

- UPDATE\s+: Matches the word "UPDATE" followed by one or more whitespace characters.
- .*: Matches any character (except for line terminators) zero or more times.
- \s+SET: Matches one or more whitespace characters followed by the word "SET".

Example 5: Finding Table Names in DDL Statements

You want to extract table names from CREATE TABLE statements in your SQL scripts.

```SQL
CREATE\s+TABLE\s+(\w+)
```

Explanation for example 5

- CREATE\s+TABLE\s+: Matches the words "CREATE TABLE" followed by one or more whitespace characters.
- (\w+): Matches one or more word characters (alphanumeric and underscore) and captures them for extraction.

For more examples, visit Regular Expressions in Visual Studio

Related content

- Search and Replace
- Search Documents Interactively
- Search Documents Using Results Lists

Feedback

Was this page helpful? Yes No

Find and Replace

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use the **Find and Replace** dialog box to locate text within a file and optionally replace it. Versions of the **Find and Replace** dialog box with slightly different options can appear, depending on how the dialog box was opened. On the **Edit** menu, point to **Find and Replace**, and then click **Quick Find** to open the dialog box with find options, but without replace options. On the **Edit** menu, point to **Find and Replace**, and then click **Quick Replace** to open the dialog box with both find options and replace options.

Toolbar buttons and shortcut keys are also available to open the **Find and Replace** dialog box.

Find What

These controls enable you to specify the string or expression that will be matched.

Find what
Type the text to search for. The dialog box attempts to fill in a probable search text, using text selected with the cursor before opening the dialog box, or nearby text, or previously searched-for text. You can reuse one of the last 20 search strings by selecting it from this drop-down list.

[string with wildcards]
If you want to use wildcards such as asterisks (`*`) and question marks (`?`) in your search string, select the **Use** check box under **Find Options** and then click **Wildcards**.

[regular expression]
To cause the search engine to interpret your search string as a regular expression, select the **Use** check box under **Find Options** and then click **Regular expressions**.

Expression Builder
This triangular button next to the **Find what** box becomes available when the **Use** check box is selected in **Find Options**. Click this button to display a list of wildcards or regular expressions, depending upon the **Use** option selected. Choosing any item from this list adds it to the string specified in the **Find what** box.

Replace With

These controls enable you to specify what will be inserted in the place of the matched string or expression.

Replace with
To replace instances of the string specified in **Find what** with another string, enter the replacement string in this field. To delete instances of the text specified in **Find what**, leave this field blank. Select the drop-down list to display the last 20 items entered. To include regular expressions in the string specified in the **Replace with** box, click the **Use** check box and then click **Regular Expressions**. This box only appears if this dialog box was opened by clicked **Quick Replace**.

Replace with
To replace instances of the string specified in the **Find what** box with another string, enter the replacement string in this field. To delete instances of the string specified in the **Find what** box, leave this field blank. Select the drop-down list to display the last 20 items entered. To include regular expressions in the string specified in the **Replace with** box, click the **Use** check box and then click **Regular Expressions**.

Expression Builder
This triangular button next to the **Replace with** box becomes available when the **Use** check box is selected in **Find Options**. Click this button to display a list of wildcards or regular expressions, depending upon the **Use** option selected. Clicking any item in this list adds it to the string specified in the **Replace with** box.

Replace
Click this button to replace the current instance of the string specified in the **Find what** box with the string specified in the **Replace with** box, and find the next instance within the scope specified in **Look in**.

Replace all
Click this button to replace all instances of the string specified in the **Find what** box with the string specified in the **Replace with** box, in all files within the scope specified in the **Look in** box.

> **⊗ Caution**
>
> Make sure that **Look in** is set to include only those files that you want to modify.

A reminder is displayed that includes a **Keep modified files open** option. To retain the **Undo** option, you must select this option. **Undo** is only available in files that remain open for editing after they are modified.

Skip File
Becomes available when the value specified for **Look in** includes multiple files. Click this button if you do not want to search or modify the current file. The search will continue in the next file on the list in **Look in**.

Look In

Look in
Select the location to look for the text specified in **Find what**. Options are **Current Document**, which searches the document window that had focus when the dialog box was opened, and **All Open Documents**, which searches all document windows that are currently open in SQL Server Management Studio.

Find Options

You can expand or collapse the **Find Options** section. The following options can be selected or cleared.

Match case
When this check box is selected, the Find Results windows will only display instances of the string specified in **Find what** that are matched both by content and by case. For example, a search for "**MyObject**" with the **Match case** check box selected will return "MyObject" but not "myobject" or "MYOBJECT."

Match whole word
When selected, the Find Results windows will only display instances of the string specified in **Find what** that are matched in complete words. For example, a search for **MyObject** will return "MyObject" but not "CMyObject" or "MyObjectC."

Search up
Search from the cursor location towards the beginning of the document.

Search hidden text
Locate instances of the text that are concealed and collapsed text.

Use
Indicates how to interpret special characters entered in the **Find what** or **Replace with** text boxes. The options include **Wildcards** and **Regular Expressions**.

Regular Expressions
Special notations define patterns of text to match. For a list, see Search Text with Regular Expressions.

Wildcards
Special characters such as asterisks (`*`) and question marks (`?`) represent one or more characters. For a list, see Search text with regular expressions.

Find Next
Begins searching for the text in the **Find what** box.

Replace
Click this button to replace the current instance of the string specified in **Find what** with the string specified in **Replace with**, and find the next instance within the scope specified in **Look in**.

Replace All
Choose this button to replace all instances of the string specified in **Find what** with the string specified in **Replace with**, in all files within the scope specified in **Look in**.

> ⊗ **Caution**
>
> Make sure that **Look in** is set to include only those files that you want to modify.

Find and Replace Views

The tabs at the top of the Find and Replace window include **View** menus. These menus enable you to choose a set of fields to display in the active pane. You can leave the **Find and Replace** window docked in a convenient location, and then change from tab to tab and view to view to perform any type of find or replace operation.

Quick Find
This toolbar tab changes the dialog box to a **Quick Find** dialog box.

Find in Files
This toolbar tab changes the dialog box to a **Find in Files** dialog box.

Quick Replace
This toolbar tab changes the dialog box to a **Quick Replace** dialog box

Replace in Files
This toolbar tab changes the dialog box to a **Replace in Files** dialog box

See Also

SQL Server Management Studio Keyboard Shortcuts

Feedback

Was this page helpful? Yes No

Find in Files

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The **Find in Files** tab of the Find and Replace window enables you to search the code of a specified set of files for a string or expression. The matches found and actions taken are listed in the Find Results window selected in **Result Options**.

Toolbar buttons and shortcut keys are also available to open the **Find and Replace** dialog box.

The sections that follow list controls available on the **Find in Files** tab.

Find What

These controls on the **Find in Files** tab enable you to specify the string or expression that will be matched.

Find what
Type the text to search for. The dialog box attempts to fill in a probable search text, using text selected with the cursor before opening the dialog box, or nearby text, or previously searched-for text. You can reuse one of the last 20 search strings by selecting it from this drop-down list.

[string with wildcards]
If you want to use wildcards such as asterisks (`*`) and question marks (`?`) in your search string, select the **Use** check box under **Find Options** and then click **Wildcards**.

[regular expression]
To cause the search engine to interpret your search string as a regular expression, select the **Use** check box under **Find Options** and then click **Regular expressions**.

Expression Builder
This triangular button next to the **Find what** box becomes available when the **Use** check box is selected in **Find Options**. Click this button to display a list of wildcards or regular expressions, depending upon the **Use** option selected. Choosing any item from this list adds it to the **Find what** string.

Look In

The option chosen from the **Look in** drop-down list determines whether **Find in Files** searches only in currently active files or in all files stored within certain folders. Select a search scope from the list, type a folder path, or click the **Browse** button to display the **Custom Directory Set Dialog Box** and choose a set of folders to search.

> ⓘ **Note**
>
> If the **Look in** option selected causes you to search a file that you have checked out from source code control, only the version of that file which has been downloaded to your local computer is searched.

Look in
Select a predefined search scope from this list, or use the **Custom Directory Set** dialog box to enter your own set of directories.

Current Document
This option is available when a document is open in an editor. It searches only the active document for the string in **Find what**.

All Open Documents
Searches all files currently opened for editing.

Current Project
Searches all files in the active project.

Entire Solution
Searches all files in the active solution.

Include subfolders
Specifies that subfolders of the folder specified in **Look in** will be searched. It requires a custom directory set.

Browse
Click this button to display the **Custom Directory Set** dialog box, where you can assemble, edit, save, and select named sets of directories to enter in the **Look in** box.

Find Options

You can expand or collapse the **Find Options** section. The following options can be selected or cleared.

Match case
When this check box is selected, the Find Results windows will only display instances of

the string specified in **Find what** that are matched both by content and by case. For example, a search for **MyObject** with the **Match case** check box selected will return "MyObject" but not "myobject" or "MYOBJECT."

Match whole word
When this check box is selected, the Find Results windows will only display instances of the string specified in **Find what** that are matched in complete words. For example, a search for **MyObject** will return "MyObject" but not "CMyObject" or "MyObjectC."

Use
Indicates how to interpret special characters entered in the **Find what** or **Replace with** text boxes. The options include **Wildcards** and **Regular Expressions**.

Regular Expressions
Special notations define patterns of text to match. For a list, see Search Text with Regular Expressions.

Wildcards
Special characters such as asterisks (`*`) and question marks (`?`) represent one or more characters. For a list, see Search text with regular expressions.

Look at these file types
This list indicates the types of files to search through in the directories specified in **Look in**. If this field is left blank, all of the files in the directories specified in **Look in** will be searched.

```
*.[ext]; *.[ext] (manual)
```

To find files of a particular type, enter an asterisk wildcard (`*`) for the file name, followed by a period (`.`) and the file extension. To find more than one file type, enter multiple search strings separated by a semicolon (`;`).

```
*.[ext]; *.[ext] (from list)
```

Select any item in the list to enter a preconfigured search string that will find files of particular types.

Result Options

Determines the location of the results when you click **Find All**. You can expand or collapse the **Result Options** section. The following options can be selected or cleared.

Find Results 1 window
When this check box is selected, the results of the current search will be appended to the content of the Find Results 1 window. This window opens automatically to display your search results. To open this window manually, click **Other Windows** on the **View** menu and then click **Find Results 1**.

Find Results 2 window
When this check box is selected, the results of the current search will be appended to the content of the Find Results 2 window. This window opens automatically to display your search results. To open this window manually, click **Other Windows** on the **View** menu and then click **Find Results 2**.

Display file names only
Displays one entry per file containing a search match rather than one entry per search match in either the Find Results 1 or Find Results 2 window. This option is not available in SQL Server Management Studio.

Keep modified files open after Replace All
When selected, leaves open all files in which replacements have been made, so you can undo or save the changes. Memory constraints might limit the number of files that can remain open after a replace operation.

> ⊗ **Caution**
>
> You can use **Undo** only on files that remain open for editing. If this option is not selected, files that were not already open for editing will remain closed, and no **Undo** option will be available in those files.

Find and Replace Views

The tabs at the top of the Find and Replace window include **View** menus. These menus enable you to choose a set of fields to display in the active pane. You can leave the Find and Replace window docked in a convenient location and then change from tab to tab and view to view to perform any type of find or replace operation.

Switch to Quick Find
This toolbar tab changes the dialog box to a **Quick Find** dialog box.

Switch to Find in Files

This toolbar tab changes the dialog box to a **Find in Files** dialog box.

Switch to Find Symbols

This toolbar tab changes the dialog box to a **Find in Symbols** dialog box.

See Also

SQL Server Management Studio Keyboard Shortcuts

Feedback

Was this page helpful? Yes No

Find Results windows

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The two Find Results windows display matches found by using the **Find in Files** or **Replace in Files** tabs of the **Find and Replace** dialog. The **Result Options** command for **Find in Files** and **Replace in Files** allows you to choose the Find Results window where any matches found will be listed.

The selected Find Results window opens automatically whenever matches are found. To display a Find Results window manually, click **Other Windows** on the **View** menu and then click **Find Results 1** or **Find Results 2**.

To display the code file and jump to the line where a match occurs, double-click any line in the results list. The source file is displayed in the Code Editor with the insertion point placed where the matched text begins. A symbol appears in the indicator margin of the editor to mark the line that includes the match, and the status bar displays its full text.

Toolbar Buttons

The following toolbar buttons are available to help you scan through the results list and jump to lines in your code where matches were found.

page flag + up arrow
Go to the line where the selected match was found.

page + left arrow
Go to the line of the previous match.

page + right arrow
Go to the line of the next match.

Clear all
Remove all matches from the **Results** list.

Shortcut Keys

The following shortcut keys are available to help you navigate quickly through the matches found.

CTRL+HOME
Scroll to the top of the Find Results window.

CTRL+END
Scroll to the bottom of the Find Results window.

PAGE UP
Scroll up to the next group of matches.

PAGE DOWN
Scroll down to the next group of matches.

UP ARROW
Select the previous match.

DOWN ARROW
Select the next match.

Search Result Entries

Each entry in the results list provides the following information:

- Full path
- File name
- Line number
- The full text of the line containing the match

> **Tip**
>
> You can use **Quick Find** to scan through a lengthy results list. Open and dock the Find Results window, and then click the triangular **View** button on the **Find** tab and switch to **Quick Find**. Set the **Look in** field of the search to **Active Window**, enter a **Find what** string, and then click **Find Next**. This lets you scan the results list for matches found in particular folders or files, or matches on lines of code where some other key term also occurs.

Summary Lines

Each set of search results begins with a line that states the search parameters, and ends with a line of statistics. For example, the results list from a **Find in Files** search in all open

documents for any strings matched by the regular expression "`var[1-3]&par`" might begin with this line of search parameters:

```
Find all "var[1-3]&par" Regular Expression, All Open Documents
```

and might conclude with this line of statistics:

```
Total found: 57 Matching files: 23 Total files searched: 59
```

The results list from a **Replace in Files** search in all open documents that replaced any strings matched by the regular expression "`var[1-3]&par`" with the string "`sample`" might begin with this line of search parameters:

```
Replace "var[1-3]&par", "sample", Regular Expression, All Open Documents
```

and might conclude with this line of statistics:

```
Total replaced: 57 Matching files: 23 Total files searched: 59
```

Feedback

Was this page helpful? Yes No

Replace in Files

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The **Replace in Files** tab of the Find and Replace window enables you to search the code of a specified set of files for a string or expression and change some or all of the matches found. The matches found and actions taken are listed in the Find Results window selected in **Result Options**.

Toolbar buttons and shortcut keys are also available to open the **Find and Replace** dialog box.

Find What

These controls on the **Replace in Files** tab enable you to specify the string or expression that will be matched.

Find what
Type the text to search for. The dialog box attempts to fill in a probable search text, using text selected with the cursor before opening the dialog box, or nearby text, or previously searched-for text. You can reuse one of the last 20 search strings by selecting it from this drop-down list.

[string with wildcards]
If you want to use wildcards such as asterisks (`*`) and question marks (`?`) in your search string, select the **Use** check box under **Find Options** and then click **Wildcards**.

[regular expression]
To cause the search engine to interpret your search string as a regular expression, select the **Use** check box under **Find Options** and then click **Regular expressions**.

Expression Builder
This triangular button next to the **Find what** box becomes available when the **Use** check box is selected in **Find Options**. Click this button to display a list of wildcards or regular expressions, depending upon the **Use** option selected. Choosing any item from this list adds it to the string specified in the **Find what** box.

Replace With

These controls enable you to specify what will be inserted in the place of the matched string or expression.

Replace with
To replace instances of the string specified in **Find what** with another string, enter the replacement string in this field. To delete instances of the string specified in **Find what**, leave this box blank. Select the drop-down list to display the last 20 items entered. To include regular expressions in the string specified in the **Replace with** box, click the **Use** check box and then click the **Regular Expressions** option.

Expression Builder
This triangular button next to the **Replace with** box becomes available when the **Use** check box is selected in **Find Options**. Click this button to display a list of wildcards or regular expressions, depending upon the **Use** option selected. Clicking any item in this list adds it to the string specified in the **Replace with** box.

Replace
Click this button to replace the current instance of the string specified in **Find what** with the string specified in the **Replace with** box, and find the next instance within the scope specified in **Look in**.

Replace all
Click this button to replace all instances of the string specified in **Find what** with the string specified in the **Replace with** box, in all files within the scope specified in **Look in**.

> ⊗ **Caution**
>
> Make sure that **Look in** is set to include only those files that you want to modify.

A reminder is displayed that includes a **Keep modified files open** option. To retain the **Undo** option, you must select this option. **Undo** is only available in files that remain open for editing after they are modified.

Skip File
Becomes available when **Look in** includes multiple files. Click this button if you do not want to search or modify the current file. The search will continue in the next file on the list in **Look in**.

Look In

The option chosen from the **Look in** drop-down list determines whether **Replace in Files** searches only in currently active files or searches all files stored within certain folders.

Select a search scope from the list, type a folder path, or click the **Browse** button to display the **Custom Directory Set** dialog box and choose a set of folders to search.

> ⓘ Note
>
> If the **Look in** option selected causes you to search a file that you have checked out from source code control, only the version of that file which has been downloaded to your local computer is searched.

Look in
Select a predefined search scope from this list, or use the **Custom Directory Set** dialog box to enter your own set of directories.

Current Document
This option is available when a document is open in an editor. Searches only the active document for the string specified in **Find what**.

All Open Documents
Searches all files currently opened for editing.

Current Project
Searches all files in the active project.

Entire Solution
Searches all files in the active solution.

Include subfolders
Specifies that subfolders of the folder specified in **Look in** will be searched. It requires a custom directory set.

Browse (...)
Click this button to display the **Choose Search Folders** dialog box, where you can assemble, edit, save, and select named sets of directories to enter in the **Look in** box.

Find Options

You can expand or collapse the **Find Options** section. The following options can be selected or cleared.

Match case
When this check box is selected, the Find Results windows will only display instances of the string specified in **Find what** that are matched both by content and by case. For

example, a search for **MyObject** with the **Match case** check box selected will return "MyObject" but not "myobject" or "MYOBJECT".

Match whole word
When this check box is selected, the Find Results windows will only display instances of the string specified in **Find what** that are matched in complete words. For example, a search for **MyObject** will return "MyObject" but not "CMyObject" or "MyObjectC."

Use
Indicates how to interpret special characters entered in the **Find what** or **Replace with** text boxes. The options include **Wildcards** and **Regular Expressions**.

Regular Expressions
Special notations define patterns of text to match. For a list, see Search Text with Regular Expressions.

Wildcards
Special characters such as asterisks (`*`) and question marks (`?`) represent one or more characters. For a list, see Search text with regular expressions.

Look at these file types
This list indicates the types of files to search through in the directories specified in **Look in**. If this box is left blank, all of the files in the directories specified in **Look in** will be searched.

```
*.[ext]; *.[ext] (manual)
```

To find files of a particular type, enter an asterisk wildcard (`*`) for the file name, followed by a period (`.`) and the file extension. To find more than one file type, enter multiple search strings separated by a semicolon (`;`).

```
*.[ext]; *.[ext] (from list)
```

Select any item in the list to enter a preconfigured search string that will find files of particular types.

Result Options

You can expand or collapse the **Result Options** section. The following options can be selected or cleared.

Find Results 1 window
When this check box is selected, the results of the current search will be appended to the content of the Find Results 1 window. This window opens automatically to display your search results. To open this window manually, click **Other Windows** on the **View** menu and then click **Find Results 1**.

Find Results 2 window
When this check box is selected selected, the results of the current search will be appended to the content of the Find Results 2 window. This window opens automatically to display your search results. To open this window manually, click **Other Windows** on the **View** menu and then click **Find Results 2**.

Display file names only
Displays one entry per file containing a search match rather than one entry per search match in either the Find Results 1 or Find Results 2 window. This option is not available in SQL Server Management Studio.

Keep modified files open after Replace All
When selected, leaves open all files in which replacements have been made, so you can undo or save the changes. Memory constraints might limit the number of files that can remain open after a replace operation.

> ⊗ **Caution**
>
> You can use **Undo** only on files that remain open for editing. If this option is not selected, files that were not already open for editing will remain closed, and no **Undo** option will be available in those files.

Find and Replace Views

The tabs at the top of the Find and Replace window include **View** menus. These menus enable you to choose a set of fields to display in the active pane. You can leave the Find and Replace window docked in a convenient location, and then change from tab to tab and view to view to perform any type of find or replace operation.

Switch to Quick Find
This toolbar tab changes the dialog box to a **Quick Find** dialog box.

Switch to Find in Files

This toolbar tab changes the dialog box to a **Find in Files** dialog box.

Switch to Find Symbols

This toolbar tab changes the dialog box to a **Find in Symbols** dialog box.

See Also

SQL Server Management Studio Keyboard Shortcuts

Feedback

Was this page helpful? Yes No

Manage Bookmarks

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

While you are working in a code editor, the **Bookmarks** window allows you to create links to specific lines of code within your document. You can display this window from the **View** menu.

To create and navigate through bookmarks, click the buttons located on the **TextEditor** toolbar and at the top of the **Bookmarks** window. You can add and remove bookmarks, activate or disable bookmarks, and organize bookmarks into folders. Certain commands are also available from the **Bookmarks** window shortcut menu. To add or remove a bookmark, place the insertion point on the desired line in the Editor, and then click **Toggle a bookmark**. To activate a bookmark, select its check box in the **Bookmarks** window; to disable (but not remove) a bookmark, clear its check box.

Text Editor Toolbar

The following buttons are enabled on the **Text Editor** toolbar when a text document is opened in the Editor. To display the **Text Editor** toolbar when you are in the Query Editor, on the **View** menu, point to **Toolbars**, and then click **Text Editor**.

Toggle a bookmark on the current line
Adds or removes a bookmark on the selected line of the document in the active Editor. Does not alter the line of code bookmarked.

Move the caret to previous bookmark
Selects the previous bookmark that is enabled in the **Bookmarks** window. When the first bookmark is reached, jumps ahead to the last one. As needed, opens the file where the selected bookmark occurs in the Editor. Scrolls that document to the bookmarked line, and places the insertion point there.

Move the caret to the next bookmark
Selects the next bookmark that is enabled in the **Bookmarks** window. When the last bookmark is reached, jumps back to the first one. As needed, opens the file where the selected bookmark occurs in the Editor. Scrolls that document to the bookmarked line, and places the insertion point there.

Clear all bookmarks in the current document
Displays a confirmation message, and then removes all bookmarks from the active document. Does not remove the lines of code that were bookmarked.

> ⊗ **Caution**
>
> You cannot undo this procedure. Afterward, you must use **Toggle a bookmark on the current line** to create new bookmarks. To disable bookmarks but not remove them, clear their check boxes in the **Bookmarks** window.

Bookmarks Window

To organize bookmarks, create bookmark folders in the **Bookmarks** window. Drag and drop bookmarks into the folders. The following buttons are available at the top of the **Bookmarks** window.

Toggle a bookmark on the current line.
Adds or removes a bookmark on the selected line of the document in the active Editor. Does not alter the line of code bookmarked.

New folder
Adds a folder to the **Bookmarks** window.

> 💡 **Tip**
>
> In a lengthy code file, it can be helpful to organize bookmarks into task-related folders. Selecting a folder enables the **Go to previous bookmark in folder** and **Go to next bookmark in folder** buttons.

Move the caret to previous bookmark
Selects the previous bookmark that is enabled in the **Bookmarks** window. When the first bookmark is reached, jumps ahead to the last one. As needed, opens the file where the selected bookmark occurs in the Editor. Scrolls that document to the bookmarked line, and places the insertion point there.

Move the caret to the next bookmark
Selects the next bookmark that is enabled in the **Bookmarks** window. When the last bookmark is reached, jumps back to the first one. As needed, opens the file where the selected bookmark occurs in the Editor. Scrolls that document to the bookmarked line, and places the insertion point there.

Move the caret to previous bookmark in the current folder
Selects the previous bookmark that is enabled within the same folder in the **Bookmarks** window. When the first bookmark is reached, jumps ahead to the last one in that folder. As needed, opens the file where the selected bookmark occurs in the Editor. Scrolls that document to the bookmarked line, and places the insertion point there.

Move the caret to next bookmark in the current folder
Selects the next bookmark that is enabled within the same folder in the **Bookmarks** window. When the first bookmark is reached, jumps back to the first one in that folder. As needed, opens the file where the selected bookmark occurs in the Editor. Scrolls that document to the bookmarked line, and places the insertion point there.

Disable/Enable All Bookmarks
Clears or enables the check boxes for all bookmarks in the **Bookmarks** window. Does not remove bookmarks, or alter the lines of code that they mark.

Delete
Removes the currently selected bookmark from the **Bookmarks** window and from the document where the bookmark occurred. Does not remove the line of code that was bookmarked.

Bookmark check boxes
Each bookmark has its own check box. To activate an existing bookmark, select its check box in the **Bookmarks** window. To hide (but not remove) an existing bookmark, clear its check box in the **Bookmarks** window.

Bookmarks Window Shortcut Menu

When you right-click on an entry in the **Bookmarks** window, the following commands are available from the shortcut menu.

Delete
Removes the currently selected bookmark from the **Bookmarks** window and from the document where the bookmark occurred. Does not remove the line of code that was bookmarked.

Rename
Allows you to assign a new display name to a bookmark or folder.

Disable/Enable Bookmark
Clears or enables the check box for the selected bookmark in the **Bookmarks** window. Does not remove the bookmark, or alter the line of code that it marks.

Disable/Enable All Bookmarks

Clears or enables the check boxes for all bookmarks in the **Bookmarks** window. Does not remove bookmarks, or alter the lines of code that they mark.

See Also

SQL Server Management Studio Keyboard Shortcuts

Feedback

Was this page helpful? Yes No

Bookmarks F1 Help

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This section contains the F1 Help for the **Bookmarks** pages in SQL Server Management Studio.

Manage Bookmarks

Feedback

Was this page helpful? Yes No

Print Code and Results

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

When you print a query or file, colorized elements print in color if your printer supports color printing and color printing is enabled in the printer setup.

However, the font and coloration of text in the Editor and the way that the text prints on the printer can differ. From the **Tools** menu, select **Options**. Under **Environment**, click **Fonts and Colors**, and then select fonts and colors for items to be displayed in the Editor and fonts and colors that take effect when printing. Because these settings can differ, the output might differ from what you see displayed in the Editor.

If you have line numbers enabled in the Editor, the line numbers will not print unless you select **Include line numbers** in the **Print** dialog box on the **File** menu.

See Also

Change Font Color, Size, and Style

Feedback

Was this page helpful? Yes No

IntelliSense (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance

The editors in SQL Server Management Studio support Microsoft IntelliSense options that reduce typing, provide quick access to syntax information, or make it easier to view the delimiters of complex expressions.

Benefits of IntelliSense

Microsoft IntelliSense provides an array of options that make language references easily accessible. When coding, you do not need to leave the editor to perform searches on language elements. You can keep your context, find the information you need, insert language elements directly into your code, and even have IntelliSense complete your typing for you.

IntelliSense Tasks

Expand table

Task Description	Topic
Describes how to configure IntelliSense options such as statement completion options or the size of Transact-SQL scripts for which you want IntelliSense turned off.	Configure IntelliSense (SQL Server Management Studio)
Describes how to use Parameter Info to get information about the number, names, and sizes of parameters for a function or stored procedure.	Parameter Info (IntelliSense)
Describes how to use Quick Info to get tooltips that describe an identifier (such as a table or view name).	Quick Info (IntelliSense)
Describes how to have IntelliSense complete the rest of an identifier after you have typed in enough of the name to get a short list of objects whose names start with the same string.	Complete Word (IntelliSense)
Describes how IntelliSense can identify both ends of a pair of delimiters, and how you can jump between the two ends of the pair.	Automatic Matching of Syntax Pairs

Task Description	Topic
Describes the conditions under which IntelliSense may not work.	Troubleshooting IntelliSense (SQL Server Management Studio)

See Also

Query and Text Editors (SQL Server Management Studio)

Feedback

Was this page helpful? Yes No

Configure IntelliSense (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance

Most Microsoft IntelliSense options are on by default. You can turn off an IntelliSense option and instead invoke it through a menu command or keystroke combination.

> **Important**
>
> Some changes will not take effect in your current editor session. You must open a new Transact-SQL editor session to see the change.

To turn statement completion options off by default

> **Note**
>
> Azure Synapse Analytics does not support IntelliSense.

1. On the **Tools** menu, click **Options**.
2. Expand **Text Editor**, expand either **All Languages**, **Transact-SQL**, or **XML**, and then click **General**.
3. Clear the check boxes for the Statement completion options that you do not want, and then click **OK**.

To modify Transact-SQL IntelliSense options

1. On the **Tools** menu, click **Options**.
2. Expand **Text Editor**, expand **Transact-SQL**, and then click **IntelliSense**.
3. Clear the check boxes for the IntelliSense options that you do not want.
4. To change the script size at which IntelliSense features are disabled, select a size from the **Maximum script size** list.

5. To change the casing applied to function names in completion lists, select a casing specification from the **Casing for built-in function names** list.

6. Select **OK**.

Feedback

Was this page helpful? Yes No

Parameter Info (IntelliSense)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The Microsoft IntelliSense **Parameter Info** option opens a parameters list that provides information about the number, names, and types of the parameters that are required by a function or stored procedure. The parameter in bold indicates the next parameter that is required as you type a function or stored procedure.

The parameter list is also displayed for nested functions. If you type a function as a parameter to another function, the parameter list displays the parameters for the inner function. Then, when the inner function parameter list is complete, the parameter list reverts to displaying the outer function parameters.

To view Parameter Info for functions or stored procedures

1. After the name of a function, type an open parenthesis as you usually type to open the parameters list. After you type the name of a stored procedure, type a space as you usually type to obtain information about the procedure parameters.

   IntelliSense displays the complete declaration for the function or the parameters for a stored procedure in a pop-up window just under the insertion point. The first parameter in the list appears in bold.

2. As you type the parameters, the bold font changes to reflect the next parameter that you need to enter.

3. Press ESC at any time to close the list, or continue typing until you have completed the function.

   For a function, if you type the closing parenthesis you also close the parameter list.

To manually start Parameter Info

1. On the **Edit** menu, select **IntelliSense** and then select **Parameter Info**.

2. Press the CTRL+SHIFT+SPACE keyboard shortcut.

For more information, see Configure IntelliSense (SQL Server Management Studio).

> **Note**
>
> The **Parameter Info** option is available only for the Database Engine Query Editor and the XML Query Editor.

Feedback

Was this page helpful? Yes No

Quick Info (IntelliSense)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The Microsoft IntelliSense **Quick Info** option displays the complete declaration for any identifier in your code. When you move the mouse pointer over an identifier, its declaration is displayed in a yellow pop-up window. In SQL Server Management Studio, **Quick Info** is available in the Database Engine and XML Query Editors.

Transact-SQL Quick Info

Quick Info displays two types of information in the Database Engine Query Editor. When not in debug mode, **Quick Info** displays the expression declaration. When in debug mode, **Quick Info** instead displays the name of the expression and its current value.

In the Database Engine Query Editor, **Quick Info** is available only for those parts of the Transact-SQL syntax that IntelliSense supports. For example, if you move the mouse pointer over the identifier for an object that has a data type that IntelliSense does not support, the **Quick Info** pop-up window contains a message that states that the data type is not supported.

See Also

Transact-SQL Syntax Supported by IntelliSense

Feedback

Was this page helpful? Yes No

Complete Word (IntelliSense)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance

The Microsoft IntelliSense Complete Word option types the rest of a variable, command, or function name once you have entered enough characters to uniquely identify the term.

There are four ways to invoke the Complete Word feature:

- On the **Edit** menu, point to **IntelliSense**, and then select **Complete Word**.
- Press ALT+RIGHT ARROW.
- Right-click the text editor, and select **Complete Word**.
- Select the **Complete Word** button on the **Text Editor** toolbar.

If what you have typed has more than one possible match, invoke Complete Word to display a list of possible matches, which you can then use to find the term and insert it into your code.

Feedback

Was this page helpful? Yes No

Automatic Matching of Syntax Pairs

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Automatic matching of syntax pairs gives you immediate feedback on whether syntax elements that must be coded in pairs are correctly paired. This is known as delimiter matching in the Database Engine Query Editor, brace matching in the Analysis Services XMLA Query Editor, and parenthesis matching in the MDX and DMX editors.

Database Engine Query Editor Delimiter Matching

The Database Engine Query Editor matches the delimiters that identify the boundaries of code blocks. The matching is done in two ways:

- The editor highlights both delimiters in a pair when you finish typing the second delimiter in the pair.
- Anytime the cursor is in one of the delimiters in a pair, you can use the CTRL+] keyboard shortcut to jump to the matching delimiter.

Delimiter Pairs

Automatic delimiter matching recognizes the following sets of delimiters:

Expand table

Lead Delimiter	Closing Delimiter
(	)
BEGIN	END
BEGIN TRY	END TRY
BEGIN CATCH	END CATCH

Automatic delimiter matching does not recognize the delimiters for bracketed identifiers ([ObjectName]), or quoted identifiers ("ObjectName"). Pair matching does not match the

single quotation delimiters for string literals ('string') because color coding already gives a visual indication of whether the string has been delimited.

Delimiter Highlighting

Matching highlights both the lead and closing element of a pair of delimiters. This lets you visually identify code blocks and check for mismatched pairs of delimiters.

Delimiters are highlighted when you type the final letter that completes the pair. For example, for a BEGIN END pair where you type BEGIN first followed by END, highlighting turns on when you type the final letter in END. You do not have to type the lead delimiter followed by the closing delimiter to turn on highlighting. If you type END first, then scroll back up the script and type BEGIN, highlighting is turned on when you type the final letter in BEGIN. The final letter typed does not have to be the end letter in the delimiter. For example, you could misspell BEGIN as BEIN, when you insert the final G the BEGIN END pair is highlighted.

The delimiter pair remains highlighted until you move the cursor. The highlighting is turned off when the cursor is moved, even if the new cursor position remains in the same delimiter. You can turn the highlighting back on by deleting and retyping any letter in either member of the pair.

Analysis Services XMLA Query Editor Brace Matching

The XMLA Query Editor brace matching shows if you have closed elements by highlighting opening and closing braces. You can also use the CTRL+] keyboard shortcut to jump from one brace to the matching brace.

The XMLA Query Editor does brace matching for the following terms:

- Matching start and end tags.
- Any pair of "<" and ">" angle brackets.
- Start and end of comments.
- Start and end of processing instructions.
- Start and end of CDATA blocks.
- Start and end of DTD declarations.

- Opening and closing quotes on attributes.

MDX and DMX Editor Parenthesis Matching

The Multi-Dimensional Expressions (MDX) and Data Mining Expressions (DMX) Editors automatically match parenthesis pairs in functions.

Feedback

Was this page helpful? Yes No

Identify issues with IntelliSense - SQL Server Management Studio (SSMS)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

There are certain cases when the IntelliSense options might not work as you expect.

Conditions That Affect IntelliSense

The following conditions might affect the behavior of IntelliSense:

- There is a code error above the cursor.

  If there is an incomplete statement or other coding error above the location of the insertion point, IntelliSense may be unable to parse the code elements, and therefore will not work. You can comment out the applicable code to enable IntelliSense again.

- The insertion point is inside a code comment.

  IntelliSense options are not available when the insertion point is within a comment in your source file.

- The insertion point is inside a string literal.

  IntelliSense options are not available when the insertion point is inside the quotation marks around a string literal, for example:

  ```
  WHERE FirstName LIKE 'Patri%|'
  ```

- The automatic options are turned off.

  Many IntelliSense features work automatically by default, but you can disable any feature.

  Even when automatic statement completion is disabled, you can use an IntelliSense feature. For more information, see Configure IntelliSense (SQL Server Management Studio).

Database Engine Query IntelliSense

The following issues apply to the SQL Server Database Engine Query Editor:

- The IntelliSense functionality of the Database Engine Query Editor does not support all Transact-SQL syntax elements. Parameter help does not support the parameters in some objects, such as extended stored procedures. For more information, see Transact-SQL Syntax Supported by IntelliSense.
- IntelliSense is only available when the Database Engine Query Editor is connected to an instance of the Database Engine from SQL Server 2008 (10.0.x) or later. IntelliSense is not available when the Query Editor is connected to earlier versions of the Database Engine.
- IntelliSense is turned off in the Database Engine Query Editor when the SQLCMD mode is set on.
- IntelliSense functionality does not cover database objects created by another connection after your editor window connected to the database. If objects are missing from IntelliSense features such as completion lists, you can choose one of these three mechanisms to refresh the cache of objects for your editor window:
  - Select the **Edit** menu, select **IntelliSense**, then select **Refresh Local Cache**.
  - Use the CTRL+Shift+R keyboard shortcut.
  - Disconnect your editor window from the instance of the Database Engine and reconnect.
- Completion lists do not include database objects for which you do not have permissions. IntelliSense flags references to objects for which you do have permissions. For example, if you open a script that is written by someone else, any references to objects for which that person has permissions and you do not are flagged as incorrect.
- Completion lists might stop working if you lose the connection to the instance of the Database Engine. Reconnect to the instance.

Feedback

Was this page helpful? Yes No

Transact-SQL Syntax Supported by IntelliSense

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes the Transact-SQL statements and syntax elements that are supported by IntelliSense in SQL Server.

Statements Supported by IntelliSense

In SQL Server, IntelliSense supports only the most commonly used Transact-SQL statements. Some general Database Engine Query Editor conditions might prevent IntelliSense from functioning. For more information, see Troubleshooting IntelliSense (SQL Server Management Studio).

> **Note**
>
> IntelliSense is not available for encrypted database objects, such as encrypted stored procedures or user-defined functions. Parameter help and Quick Info are not available for the parameters of extended stored procedures and CLR Integration user-defined types.

SELECT Statement

The Database Engine Query Editor provides IntelliSense support for the following syntax elements in the SELECT statement:

SELECT

WHERE

FROM

ORDER BY

HAVING

UNION

FOR

GROUP BY

TOP

OPTION (hint)

Additional Transact-SQL Statements That Are Supported

The Database Engine Query Editor also provides IntelliSense support for Transact-SQL statements that are shown in the following table.

Expand table

Transact-SQL statement	Syntax supported	Exceptions
INSERT	All syntax, except the *execute_statement* clause.	None
UPDATE	All syntax.	None
DELETE	All syntax.	None
DECLARE @local_variable	All syntax.	None
SET @local_variable	All syntax.	None
EXECUTE	Execution of user-defined stored procedures, system stored procedures, user-defined functions, and system functions.	None
CREATE TABLE	All syntax.	None
CREATE VIEW	All syntax.	None
CREATE PROCEDURE	All syntax.	There is no IntelliSense support for the EXTERNAL NAME clause. In the AS clause, IntelliSense supports only the statements

Transact-SQL statement	Syntax supported	Exceptions
		and syntax that are listed in this topic.
ALTER PROCEDURE	All syntax	There is no IntelliSense support for the EXTERNAL NAME clause. In the AS clause, IntelliSense supports only the statements and syntax that are listed in this topic.
USE	All syntax.	None

IntelliSense in Supported Statements

IntelliSense in the Database Engine Query Editor supports the following syntax elements when they are used in one of the supported Transact-SQL statements:

- All join types, including APPLY
- PIVOT and UNPIVOT
- References to the following database objects:
  - Databases and schemas
  - Tables, views, table-valued functions, and table expressions
  - Columns
  - Procedures and procedure parameters
  - Scalar functions and scalar expressions
  - Local variables
  - Common table expressions (CTE)
- Database objects that are referenced only in CREATE or ALTER statements in the script or batch, but which do not exist in the database because the script or batch has not yet been run. These objects are as follows:
  - Tables and procedures that have been specified in a CREATE TABLE or CREATE PROCEDURE statement in the script or batch.

- Changes to tables and procedures that have been specified in an ALTER TABLE or ALTER PROCEDURE statement in the script or batch.

> **Note**
>
> IntelliSense is not available for the columns of a CREATE VIEW statement until the CREATE VIEW statement has been executed.

IntelliSense is not provided for the previously listed elements when they are used in other Transact-SQL statements. For example, there is IntelliSense support for column names that are used in a SELECT statement, but not for columns that are used in the CREATE FUNCTION statement.

Examples

Within a Transact-SQL script or batch, IntelliSense in the Database Engine Query Editor supports only the statements and syntax that are listed in this topic. The following Transact-SQL code examples show what statements and syntax elements IntelliSense supports. For example, in the following batch, IntelliSense is available for the `SELECT` statement when it is coded by itself, but not when the `SELECT` is contained in a `CREATE FUNCTION` statement.

```
USE AdventureWorks2022;
GO
SELECT Name
FROM Production.Product
WHERE Name LIKE N'Road-250%' and Color = N'Red';
GO
CREATE FUNCTION Production.ufn_Red250 ()
RETURNS TABLE
AS
RETURN
(
    SELECT Name
    FROM AdventureWorks2022.Production.Product
    WHERE Name LIKE N'Road-250%'
      AND Color = N'Red'
);GO
```

This functionality also applies to the sets of Transact-SQL statements in the AS clause of a CREATE PROCEDURE or ALTER PROCEDURE statement.

Within a Transact-SQL script or batch, IntelliSense supports objects that have been specified in a CREATE or ALTER statement; however, these objects do not exist in the database because the statements have not been executed. For example, you might enter the following code in the Query Editor:

```
USE MyTestDB;
GO
CREATE TABLE MyTable
    (PrimaryKeyCol   INT PRIMARY KEY,
    FirstNameCol      NVARCHAR(50),
   LastNameCol       NVARCHAR(50));
GO
SELECT
```

After you type `SELECT`, IntelliSense lists **PrimaryKeyCol**, **FirstNameCol**, and **LastNameCol** as possible elements in the select list, even if the script has not been executed and `MyTable` does not yet exist in `MyTestDB`.

Feedback

Was this page helpful? Yes No

Transact-SQL Code Snippets

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

A Transact-SQL.code snippet is a template containing the basic structure of a Transact-SQL statement or block. You can use snippets as a starting point when adding statements in the Database Engine Query Editor. You can insert the pre-defined snippets supplied with SQL Server, or create your own.

Transact-SQL Code Snippet Tasks

Expand table

Task Description	Topic
Describes how to insert an existing snippet, such as a CREATE TABLE or SELECT snippet.	Insert Transact-SQL Snippets
Describes how to enclose a set of Transact-SQL statements in a BEGIN, IF, or WHILE block.	Insert Surround-with Transact-SQL snippets
Describes how to finish building a complete Transact-SQL statement or block after inserting a code snippet.	Complete Transact-SQL Snippets
Describes how to build your own custom snippets and add them to the set of SQL Server snippets.	Add Transact-SQL Snippets

See Also

Query and Text Editors (SQL Server Management Studio)

Feedback

Was this page helpful? Yes No

Insert Transact-SQL Snippets

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

A Transact-SQL code snippet is a template you can use as a starting point when writing new Transact-SQL statements in the Database Engine Query Editor.

Inserting Snippets

You can use the **Insert Snippet** menu to open a categorized list of snippets to choose from.

Transact-SQL snippets contain replacement points: text that suggests the syntax relevant to that point. For example, the CREATE TABLE snippet has replacement points for elements such as the table name, column names, and column data types. After inserting the snippet, you must change the replacement text to form a valid Transact-SQL statement. For more information, see Complete Transact-SQL Snippets.

Inserting a Snippet by Using the Insert Snippet Menu

1. In the Database Engine Query Editor window, put the cursor where you want to insert the Transact-SQL snippet.
2. Launch the snippet picker tooltip in one of three ways:
   - Press CTRL+K, CTRL+X.
   - On the **Edit** menu, point to **IntelliSense**, then click **Insert Snippet**.
   - Right-click the mouse and then select the **Insert Snippet** command from the shortcut menu.
3. Double-click the snippet, or select the snippet from the snippet picker and then press TAB or ENTER.

See Also

Insert Surround-with Transact-SQL snippets

Feedback

Was this page helpful? Yes No

Insert Surround-with Transact-SQL snippets

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

A surround-with snippet is a template you can use as a starting point when enclosing a set of Transact-SQL statements in a BEGIN, IF, or WHILE block.

Inserting Surround-with Snippets

Surround-with snippets can be launched by one of three ways: through a keyboard shortcut, through the **Edit** menu, and through the context menu.

After inserting the snippet, you must change the replacement text to form a valid Transact-SQL statement. For more information, see Complete Transact-SQL Snippets.

To insert a surround-with snippet

1. In the Database Engine Query Editor window, select the set of statements to be included in the block.
2. Use one of these three methods to display the list of surround-with snippets:
   - Type CTRL+K, CTRL+S.
   - From the **Edit** menu, point to **IntelliSense**, and then select the **Surround With** command.
   - Right-click the selected text, and then select the **Surround With** command from the context menu.
3. Select the name of the snippet (BEGIN, IF, or WHILE) from the list either by using the mouse, or by typing the name of the snippet and pressing TAB or ENTER.

See Also

Insert Transact-SQL Snippets

Feedback

Was this page helpful? Yes No

Complete Transact-SQL Snippets

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Once a Transact-SQL code snippet has been inserted into a script, you edit the contents of the snippet to build a complete Transact-SQL statement.

Completing Snippets

When you add a Transact-SQL snippet to your script, the inserted snippet statement has one or more replacement points, which are highlighted. If you rest your mouse pointer on a replacement point, a tooltip appears with a description of the syntax element you can specify. The Database Engine Query Editor recognizes the snippet as separate from the surrounding script until you close the source file. The replacement points remain active until you close the source file.

You can also add additional syntax elements to the template code inserted by a snippet. For example, the Create Table snippet template generates two column definitions; you must add additional column definitions to create a table with more than two columns.

Completing the Snippet Statement

1. Use the TAB key to move from one replacement point to the next. Use SHIFT+TAB to move to the previous replacement point.
2. Click CTRL+SPACE to invoke IntelliSense.
3. Select an item from the list, or type a replacement of your choice.

See Also

Insert Transact-SQL Snippets
Insert Surround-with Transact-SQL snippets

Feedback

Was this page helpful? Yes No

Add Transact-SQL Snippets

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can add your own Transact-SQL code snippets to the set of pre-defined snippets included in SQL Server.

Creating a Transact-SQL Snippet File

The first part of creating a Transact-SQL code snippet is to create an XML file with the text of your code snippet. The file must have a .snippet file extension, and meet the requirements of the Code Snippets Schema. Set the snippet language to SQL.

You can use the pre-defined snippets that ship with SQL Server as examples. To find the pre-defined snippets, open SQL Server Management Studio, select the **Tools** menu, and click **Code Snippet Manager**. Select **SQL** in the **Language** list box, the path to the Transact-SQL snippets is displayed in the **Location** box.

Registering the Code Snippet

After creating the snippet file, use the Code Snippets Manager to register the snippet with SQL Server Management Studio. You can either add a folder containing multiple snippets, or import individual snippets to the **My Code Snippets** folder.

Procedures

Adding a Snippet Folder

1. Open SQL Server Management Studio.
2. Select the **Tools** menu, and click **Code Snippets Manager**.
3. Click the **Add** button.
4. Navigate to the folder containing your code snippets, and click the **Select Folder** button.

Importing a Snippet

1. Open SQL Server Management Studio.
2. Select the **Tools** menu, and click **Code Snippets Manager**.
3. Click the **Import** button.
4. Navigate to the folder containing your snippet, click on the .snippet file, and click the **Open** button.

Examples

The following example creates a **TRY-CATCH** surround-with snippet and imports it to SQL Server Management Studio.

1. Paste the following code into notepad, then save as a file named TryCatch.snippet.

```
<?xml version="1.0" encoding="utf-8" ?>
<CodeSnippets
xmlns="http://schemas.microsoft.com/VisualStudio/2005/CodeSnippet">
<_locDefinition xmlns="urn:locstudio">
    <_locDefault _loc="locNone" />
    <_locTag _loc="locData">Title</_locTag>
    <_locTag _loc="locData">Description</_locTag>
    <_locTag _loc="locData">Author</_locTag>
    <_locTag _loc="locData">ToolTip</_locTag>
   <_locTag _loc="locData">Default</_locTag>
</_locDefinition>
<CodeSnippet Format="1.0.0">
<Header>
<Title>TryCatch</Title>
                        <Shortcut></Shortcut>
<Description>Example Snippet for Try-Catch.</Description>
<Author>SQL Server Books Online Example</Author>
<SnippetTypes>

<SnippetType>SurroundsWith</SnippetType>
</SnippetTypes>
</Header>
<Snippet>
<Declarations>
                                <Literal>
                                <ID>CatchCode</ID>
                                <ToolTip>Code to handle the caught
error</ToolTip>
                                <Default>CatchCode</Default>
                                </Literal>
</Declarations>
<Code Language="SQL"><![CDATA[
BEGIN TRY
```

```
$selected$ $end$

END TRY
BEGIN CATCH

$CatchCode$

END CATCH;
]]>
</Code>
</Snippet>
</CodeSnippet>
</CodeSnippets>
```

2. Open SQL Server Management Studio.
3. Select the **Tools** menu, and click **Code Snippets Manager**.
4. Click the **Import** button.
5. Navigate to the folder containing TryCatch.snippet, click on the TryCatch.snippet file, and click the **Open** button. You should now have a TryCatch snippet in your **My Code Snippets** folder.

See Also

Insert Surround-with Transact-SQL snippets

Feedback

Was this page helpful? Yes No

Status Bar (Database Engine Query Editor)

Article • 02/13/2025

Applies to: SQL Server

The status bar of Database Engine Query Editor windows can be color coded to indicate which instance of the Database Engine each window is connected to.

1. **Before you begin:** Status Bar Colors
2. **To set a server status color in:** Object Explorer, Registered Server
3. **To use a status color:** Open Query Editor Using a Server Color, Open a Query Editor Specifying a Status Color

This feature works with SSMS version 17.9.1 and earlier.

Status Bar Colors

You can associate a status bar color with a specific server node in either **Object Explorer** or **Registered Servers**. The colors can only be specified for server nodes connected to an instance of the Database Engine, not server nodes for other SQL Server technologies. You can also specify a custom status bar color each time you connect a new Database Engine Query Editor window to an instance of the Database Engine. You can then open a query editor window using either the status color defined for the server node, or specify a unique color for that editor window.

Setting a custom status bar color for a server node in Object Explorer must be done when making the connection. To change the color associated with an existing server node, you must disconnect and then reconnect specifying the new color.

Set the Status Color for a Server in Object Explorer

To set a server status color in Object Explorer

1. In **Object Explorer**, select the **Connect** button and then select **Database Engine...**.
2. On the **Connect to Server** dialog, select **Options >>**.

3. Select the **Use custom color** check box.
4. To select the color, select the **Select...** button.
5. Select either a basic or custom color, then select OK.
6. Fill in the rest of the connection information, and then select the **Connect** button.

Set the Status Color For a Registered Server

To set a server color For a Registered Server

1. In **Registered Servers**, right-click the server node and then select **Properties...**.
2. On the **Edit Server Registration Properties** dialog, select the **Connection Properties** tab.
3. Select the **Use custom color** check box.
4. To select the color, select the **Select...** button.
5. Select either a basic or custom color, then select OK.
6. Select the **Save** button on the **Edit Server Registration Properties** dialog.

Open An Editor Using a Server Color

To open an editor window using a server color

- Right-click a server node in either **Object Explorer** or **Registered Servers**, and select **New Query**.
- Alternatively, highlight a server node, and then select the **New Query** toolbar button.
- The status bar in the editor window will use the color defined for the associated server.

Open an Editor Specifying a Status Color

To open an editor window specifying a status color

- Open the **File** menu, select **New**, and then select **Database Engine Query**.
- On the **Connect to Server** dialog, select **Options >>**.

- Select the **Use custom color** check box.
- To select the color, select the **Select...** button.
- Select either a basic or custom color, then select OK.
- Fill in the rest of the connection information, and then select the **Connect** button.

See Also

Query and Text Editors (SQL Server Management Studio)

Feedback

Was this page helpful? Yes No

View Spatial Data in Object Explorer

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The **Spatial results** window in Query Editor provides visual mapping tools for viewing spatial data results in addition to the data displayed in grid format in the **Results** window. To display spatial data in the **Spatial results** window, your query results must contain at least one spatial data column with either geometry or geography data.

To view spatial data in the Spatial results window

1. In Query Editor, click the **Spatial results** tab.

   > **Note**
   >
   > This window is not available if your query results do not contain spatial data or if you specify that your results are returned as text.

2. Select the column you want to view from the **Select spatial column** list. If there is only one spatial column in your table, this column is the default option in the list.

3. Select the non-spatial column you want to use as data labels from the **Select label columns** list.

4. Select the projection you want for geography data from the **Select projection** list. The default projection is Equirectangular; other projections available are Mercator, Robinson, or Bonne.

   > **Note**
   >
   > **Select projection** is not available if your spatial column contains geometry data.

5. Adjust the **Zoom** slider to increase the visual size of mapped elements. For polygon shapes, the label is visible only when the shape is large enough to hold the label text.

See Also

Spatial Results Window
Database Engine Query Editor (SQL Server Management Studio)
Query and Text Editors (SQL Server Management Studio)

Feedback

Was this page helpful? Yes No

Spatial Results Window

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The **Spatial results** window provides visual mapping tools for viewing spatial data. To view spatial results, your query results must include a spatial column with either geometry or geography data.

> **Note**
>
> The **Spatial results** window is only available if your results are returned to a grid in the **Results** window. If you specify that your results are returned as text, this window is not available.

Options

Select spatial column
Specify the spatial column you want to view from the spatial columns in the query results. Only one column can be selected at a time.

Select label column
Specify the non-spatial column from the columns returned in the query results to label the spatial data. Only one column can be selected at a time.

This option is not available when only point instances are returned in a query.

Select projection
Display geography data in one of four projections: Equirectangular, Mercator, Robinson, or Bonne.

This option is not available for geometry data.

Zoom
Adjust the map display on an exponential scale.

Show grid lines
Toggle coordinate gridlines on or off.

For polygon shapes, the label is displayed only when the shape is large enough to hold the label text. To display labels for small shapes, adjust the zoom.

> **Note**
>
> Point instances cannot be labeled.

See Also

View Spatial Data in Object Explorer
Spatial Data (SQL Server)
Database Engine Query Editor (SQL Server Management Studio)
Query and Text Editors (SQL Server Management Studio)

Feedback

Was this page helpful? Yes No

Install non-English language versions of SQL Server Management Studio (SSMS)

Article • 02/13/2025

SSMS is available in several languages, but the SSMS installer blocks installation on computers when their system locale doesn't match the SSMS language.

> **Note**
>
> This article applies to SSMS 17.x. For SSMS 18.x the block on mixed languages setup has been removed and you can now, for example, install SSMS German on a French version of Windows. If the OS language does not match the SSMS language, set the desired language under **Tools** > **Options** > **International Settings**, otherwise SSMS will show the English UI.

The following directions differ depending on your version of Windows. The following are for Windows 10.

Install non-English SSMS on a computer running an English operating system (OS)

1. Install the Windows language pack for the language you want SSMS to use:
    - **Settings** > **Time & language** > **Region & language** > **Add a language**
2. Now set the system locale to use the language pack installed in the previous step by clicking the language just installed, then select **Set as default**. (After installing SSMS, you can set the system locale back to English.)
3. After your operating system is running in the desired language, install the SSMS language you want. The first time you install a new SSMS language, use the full package. You can use the upgrade package for subsequent installs.
4. Run SSMS, and it should display as the language you installed in the previous step.
5. Set your computer's system locale back to English.

Install SSMS in a language other than the language of the installed OS

1. Install the Windows language pack for the language you want SSMS to use:

- **Settings** > **Time & language** > **Region & language** > **Add a language**

2. Now set the system locale to use the language pack installed in the previous step by clicking the language just installed, then select **Set as default**.

3. After your operating system is running in the desired language, install the SSMS language you want. The first time you install a new SSMS language, use the full package. You can use the upgrade package for subsequent installs.

4. For each language you want to install that does not match the language of the first version of SSMS you installed, install the corresponding Visual Studio 2015 Shell (Isolated) Language Pack:

   - Browse to https://connect.microsoft.com/VisualStudio/ExtendVS (you may need to sign in and complete the *Connect Registration* process).
   - Download the desired Visual Studio 2015 Shell (Isolated) Language Pack and install it.

   > **Important**
   >
   > Use the previous steps to install the Visual Studio 2015 Isolated Shell Language Pack, do not use the **Get additional languages** link on **Tools** | **Options** | **International Settings**.

5. Run SSMS and select the language you want to use in:

   - **Tools** | **Options** | **International Settings**

6. Close and restart SSMS.

Next steps

- Tutorial: SQL Server Management Studio

Feedback

Was this page helpful? Yes No

Use templates in SQL Server Management Studio

Article • 02/13/2025

This tutorial introduces you to the prebuilt Transact-SQL (T-SQL) templates that are available in SQL Server Management Studio (SSMS). In this article, you learn how to:

Prerequisites

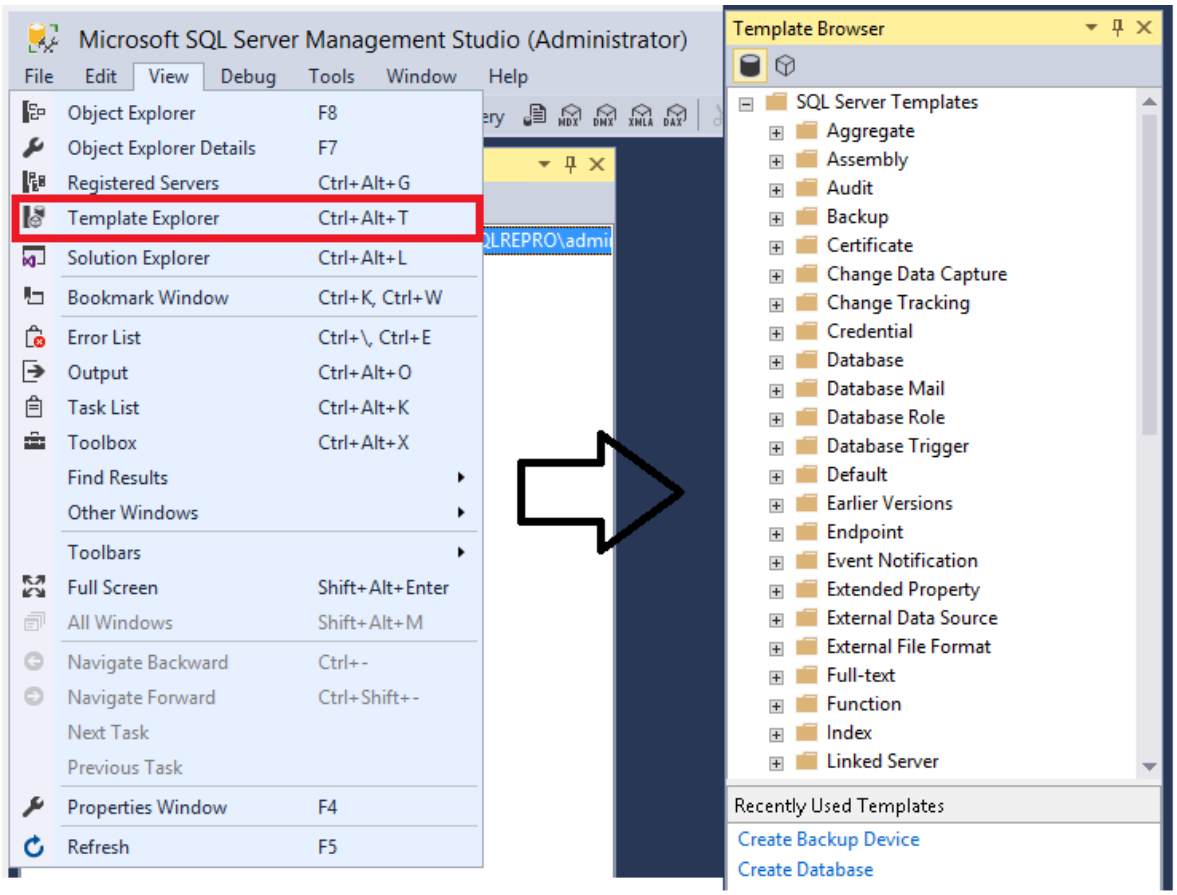
To complete this tutorial, you need SQL Server Management Studio and access to a SQL Server.

- Install SQL Server Management Studio.
- Install SQL Server 2017 Developer Edition .

Use Template Browser

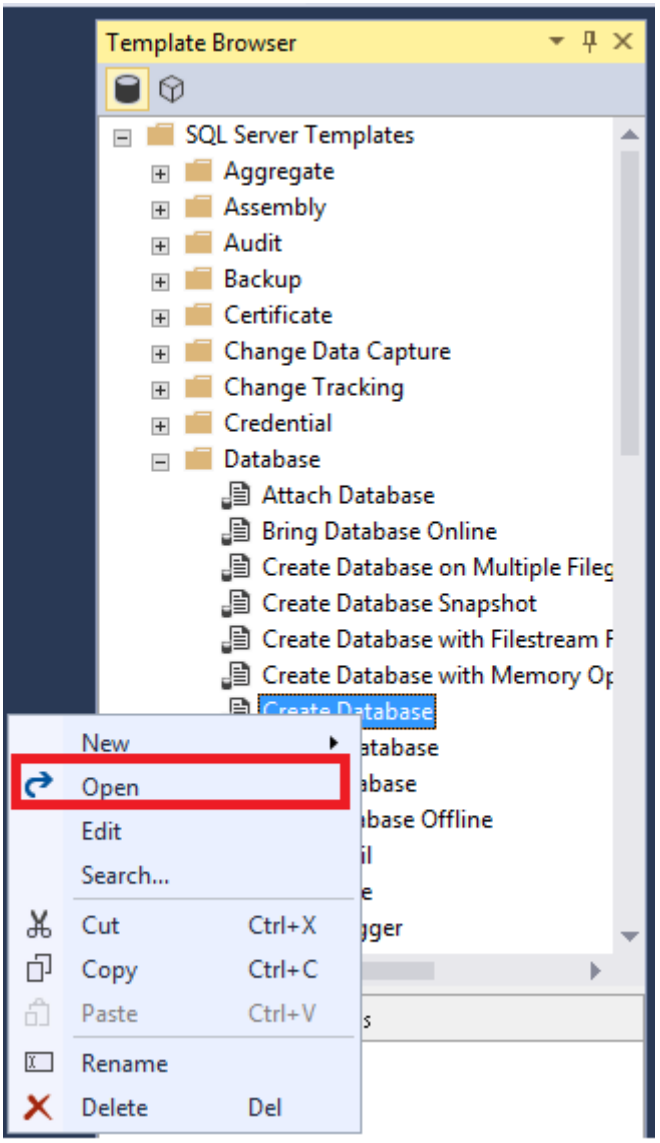
In this section, you learn how to locate and use Template Browser.

1. Open SQL Server Management Studio.
2. In the **View** menu, select **Template Browser** (Ctrl+Alt+T):



You can see recently used templates at the bottom of the template browser.

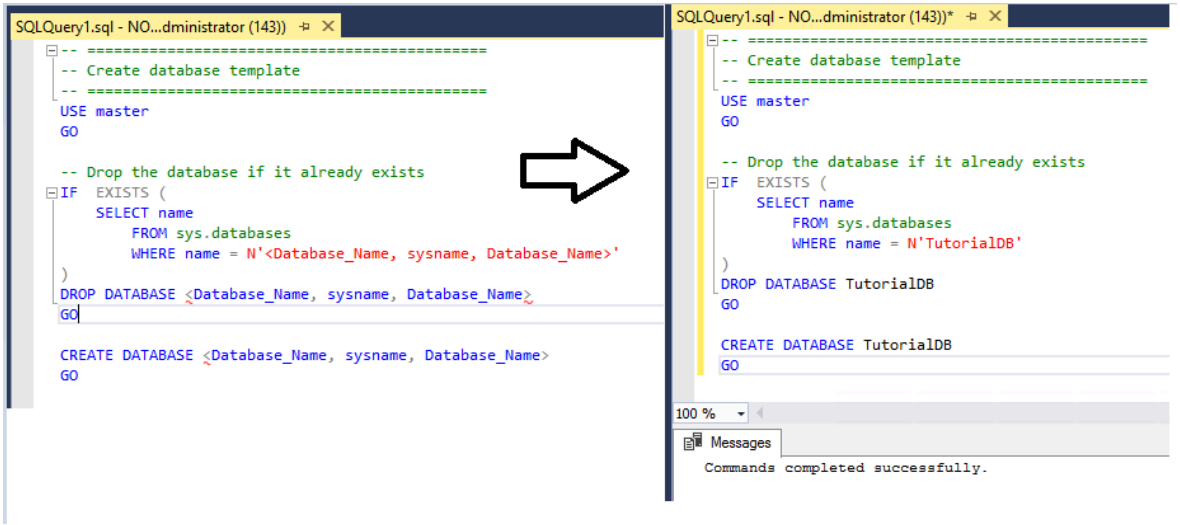
3. Expand the node you're interested in. Right-click the template, and then select **Open**:



You can also double-click the template name to open it.

4. A new query window opens. The T-SQL script is already populated.

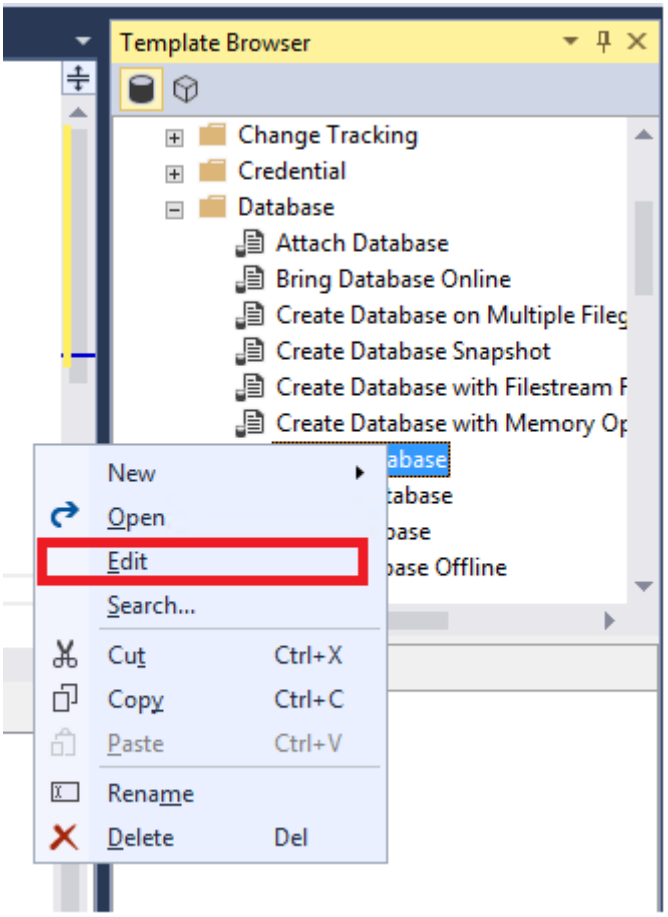
5. Modify the template to suit your needs, and then select **Execute** to run the query:



Edit an existing template

You can also edit existing templates in Template Browser.

1. In Template Browser, go to the template you want to work with.
2. Right-click the template, and then select **Edit**:

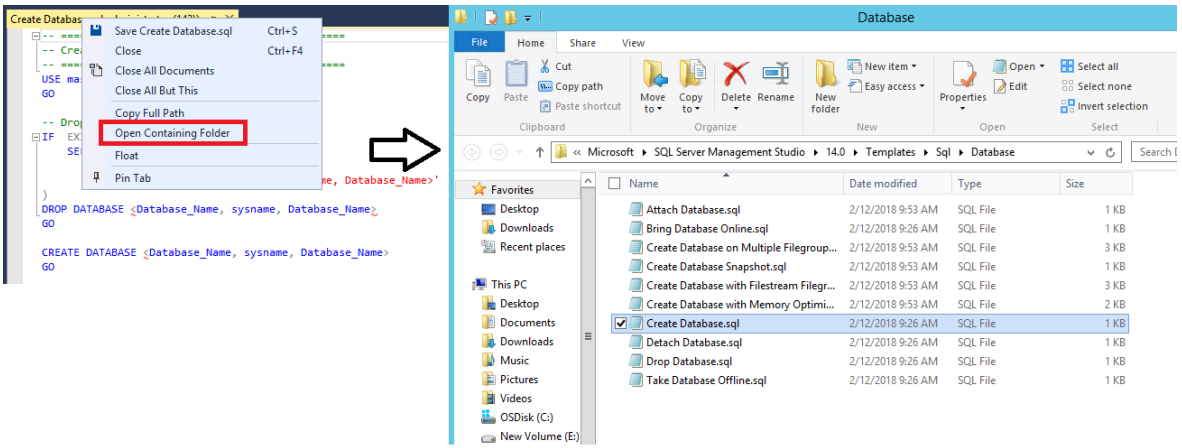


3. In the query window that opens, make the changes that you want to make.
4. To save the template, select **File** > **Save** (Ctrl+S).
5. Close the query window.
6. Reopen the template. Your edits should appear.

Locate templates on disk

When a template is open, you can locate the templates that are on disk.

1. In Template Browser, select a template, and then select **Edit**.
2. Right-click **Query Title**, and then select **Open Containing Folder**. The explorer should open where the templates are stored on disk:



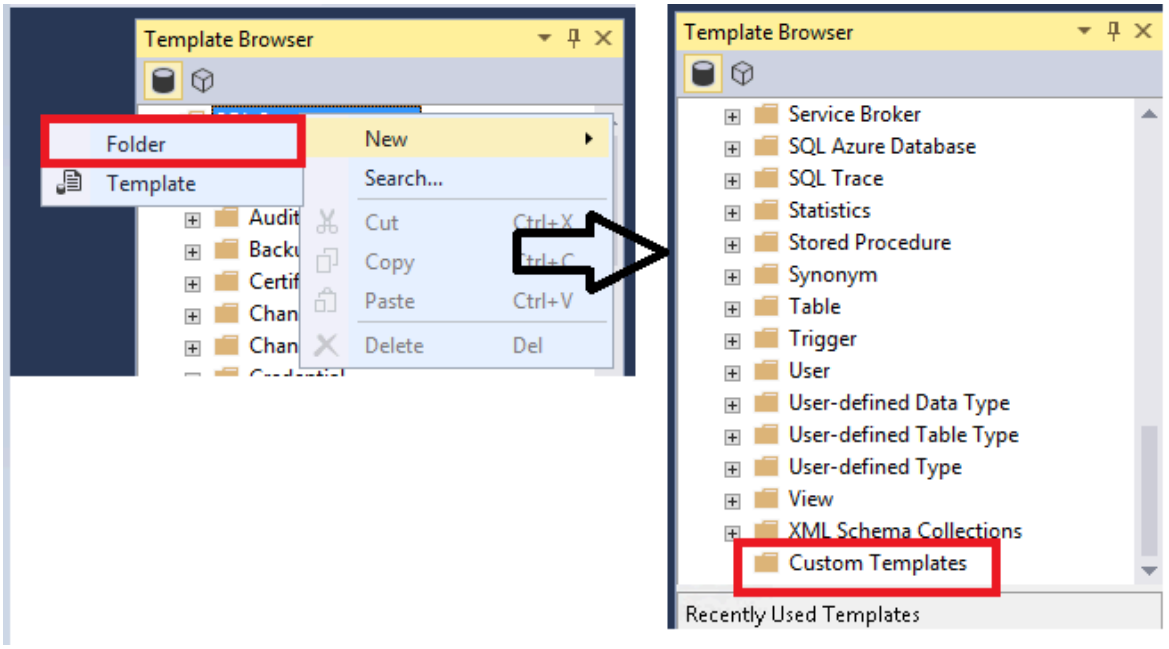
Create a new template

You can also create a new template in Template Browser. The following steps show you how to create a new folder, and then create a new template in that folder. You can also use these steps to create a custom template in an existing folder.

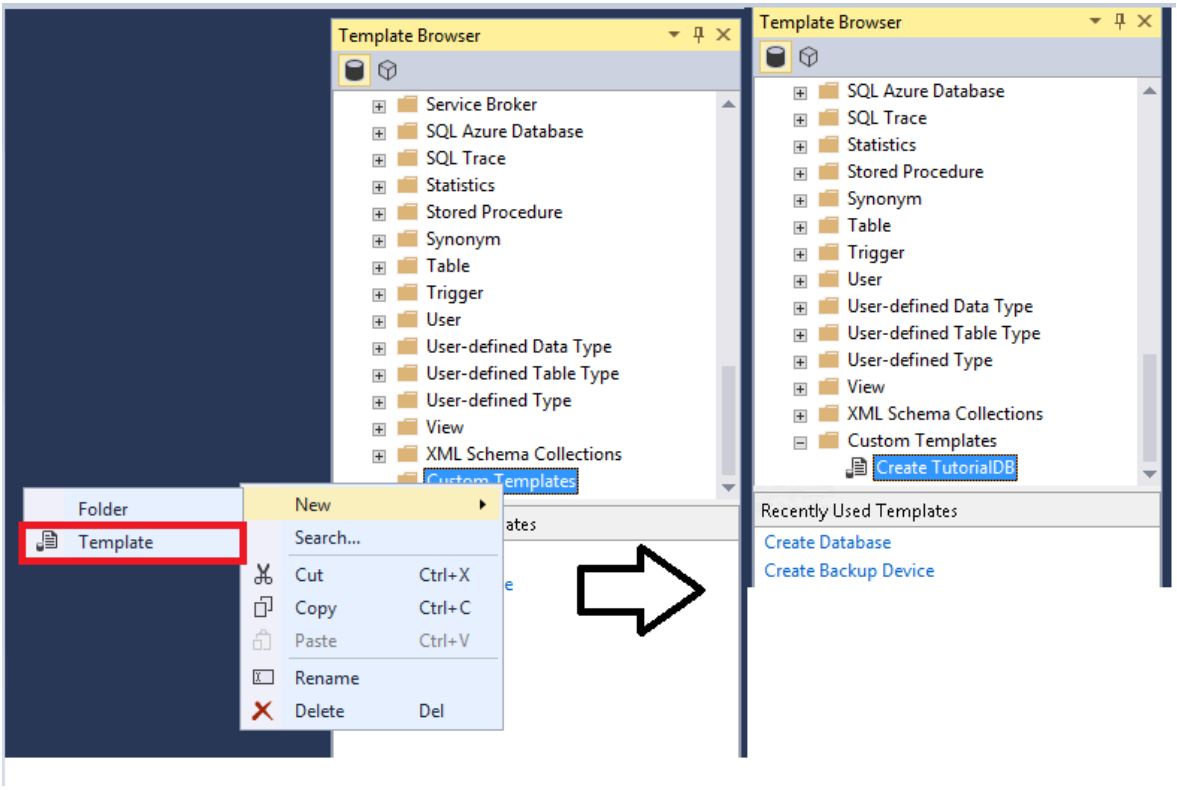
1. Open Template Browser.

2. Right-click **SQL Server Templates**, and then select **New** > **Folder**.

3. Name this folder **Custom Templates**:



4. Right-click the newly created Custom Templates folder, and then select **New** > **Template**. Enter a name for your template:



5. Right-click the template you created, and then select **Edit**. The New Query Window opens.

6. Enter the T-SQL text that you want to save.

7. In the **File** menu, select **Save**.

8. Close the existing query window, and then open your new custom template.

Next steps

The best way to get acquainted with SSMS is through hands-on practice. These *tutorial* and *how-to* articles help you with various features available within SSMS. These articles teach you how to manage the components of SSMS and how to find the features that you use regularly.

- Connect to and query an instance
- Scripting
- SSMS Configuration
- Additional Tips and Tricks for using SSMS

Feedback

Was this page helpful?

Yes No

Customize Menus and Shortcut Keys

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

A keyboard accelerator allows you to select a menu command or button by pressing ALT+<*single letter*>. For example, to open the **Edit** menu, press ALT+E. You can rearrange and modify toolbar buttons, menus, and menu commands by using the **Customize** dialog box. Instructions are provided for changing the settings using the mouse and using only the keyboard.

Keyboard accelerators for stored procedures using the Ctrl key can be created from the **Keyboard** page of the **Tools/Options** dialog box.

> **Note**
>
> Click **Collapse All** at the top of this page to show only the headings.

Opening the Keyboard Accelerator Dialog Box Using the Mouse

To access the dialog box for assigning or changing a keyboard accelerator (using the mouse)

1. On the **Tools** menu, click **Customize**.
2. Make sure the toolbar you want to change is visible.
   a. In the **Customize** dialog box, click the **Toolbars** tab.
   b. Select the check box for the toolbar you want to display.
3. In the **Customize** dialog box, click the **Commands** tab.

Changing a Toolbar Buttons Accelerator Key Using the Mouse

To assign or change a toolbar button's keyboard accelerator (using the mouse)

1. Click the button on the toolbar.
2. In the **Customize** dialog box, on the **Commands** tab, click **Modify Selection**.
3. In the **Name** box on the shortcut menu, type a name for the toolbar button with an ampersand (&) before the letter that you want as the keyboard accelerator.
4. Press **ENTER**.
5. In the **Customize** dialog box, click **Close**.

Changing a Menu Commands Accelerator Key Using the Mouse

To assign or change a menu command's keyboard accelerator (using the mouse)

1. Click the menu name on the menu bar or toolbar.
2. Click the menu command.
3. In the **Customize** dialog box, click **Modify Selection**.
4. In the **Name** box on the shortcut menu, type a name for the menu command with an ampersand (&) before the letter that you want as the keyboard accelerator.
5. Press **ENTER**.
6. In the **Customize** dialog box, click **Close**.

Opening the Keyboard Accelerator Dialog Box Using the Keyboard

To access the dialog box for assigning or changing a keyboard accelerator (using the keyboard)

1. Press ALT+T, then type C, to open the **Customize** dialog box.
2. Make sure the toolbar you want to change is visible.

a. In the **Customize** dialog box, press ALT+B to show the **Toolbars** tab.

b. Use the arrow keys to select the toolbar you want to display, then **SPACE** to select the check box.

3. In the **Customize** dialog box, press ALT+C to display the **Commands** tab.

Changing a Toolbar Buttons Accelerator Key Using the Keyboard

To assign or change a toolbar button's keyboard accelerator (using the keyboard)

1. Press ALT+R to display the **Rearrange Commands** dialog box.
2. In the **Rearrange Commands** dialog box, use the arrow keys to select **Toolbar**.
3. Tab to the **Toolbar** list, and use the arrow keys to select the toolbar that contains the button you want to change, and then press **ENTER**.
4. Tab to the **Controls** list, and use the arrow keys to select the button you want to change.
5. Press **ALT+M**, to select **Modify Selection**.
6. Tab to the **Name** box on the shortcut menu, type a name for the toolbar button with an ampersand (&) before the letter that you want as the keyboard accelerator.
7. Press **ENTER**.
8. Tab to the **Close** button, and then press **ENTER**.

Changing a Menu Commands Accelerator Key Using the Keyboard

To assign or change a menu command's keyboard accelerator (using the keyboard)

1. Press ALT+R to display the **Rearrange Commands** dialog box.
2. Tab to **Menu Bar** and then use the arrow keys to click the menu you want in the **Menu Bar** list, and then press **ENTER**.

3. Tab to the **Controls** list, and use the arrow keys to select the button you want to change.
4. Press ALT+M, to select **Modify Selection**.
5. Tab to the **Name** box on the shortcut menu, and type a name for the toolbar button with an ampersand (&) before the letter that you want as the keyboard accelerator.
6. Press **ENTER**.
7. In the **Customize** dialog box, click **Close**.

Creating a Keyboard Accelerator for a Stored Procedure

To create a keyboard accelerator for a stored procedure

1. On the **Tools** menu, click **Options**.
2. On the **Keyboard** page, select an unused keyboard combination in the **Shortcut** list.
3. In the **Stored Procedure** box, type the stored procedure name, and then click **OK**.

Adding a New Item to the Menu

To add a new item to the menu

1. On the **Tools** menu, click **Options**.
2. In the **Customize** dialog box, on the **Commands** tab, click **New Menu**.
3. On the **Commands** box, drag **New Menu** to the menu bar and drop it where you want the new menu to appear.
4. On the menu, right-click **New Menu**, and in the **Name** box, type a name for the new menu.
5. In the **Customize** dialog box, select category such as **File**, and select a command such as **Open File**. Drag the command to the new menu. As you point to the new menu, the menu will expand. Drop the command onto the expanded menu.

6. In the **Customize** dialog box, click **Close**.

> **Note**
>
> Some commands are available only when SQL Server Management Studio is displaying relevant content. If no commands on the menu are available, the menu item will not be available.

See Also

Features in SQL Server Management Studio

Feedback

Was this page helpful? Yes No

Configure WMI to Show Server Status in SQL Server Tools

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes how to configure WMI to show the server status in SQL Server tools in SQL Server. When connecting to servers, both the Registered Servers and Object Explorer components of SQL Server Management Studio, as well as SQL Server Configuration Manager, use Windows Management Instrumentation (WMI) to obtain the status of the SQL Server (MSSQLSERVER) and SQL Server Agent (MSSQLSERVER) services. To display the status of the service, the user must have rights to remotely access the WMI object. The server must have WMI installed to configure this permission.

To configure WMI permission

1. On the **Start** menu on the remote server, click **Run**.
2. In the **Open** box type **wmimgmt.msc**, and then click **OK**.
3. In the **Windows Management Infrastructure** program, right-click **WMI Control (Local)**, and then click **Properties**.
4. In the **WMI Control (Local) Properties** dialog box, on the **Security** tab, expand **Root**, and then click **CIMV2**.
5. Click **Security** to open the **Security for ROOT\CIMV2** dialog box.
6. Add a group or user to the **Group or user names** box and select it.
7. In the **Permissions for**<*group or user*> box, select the **Allow** column, for the **Remote Enable** permission, for users whom you wish to remotely detect the service status.

See Also

Start, Stop, or Pause the SQL Server Agent Service

Feedback

Was this page helpful? Yes No

Configure login auditing (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server - Windows only

This article describes how to configure login auditing in SQL Server on Windows, to monitor SQL Server Database Engine login activity. Login auditing can be configured to write to the error log on the following events.

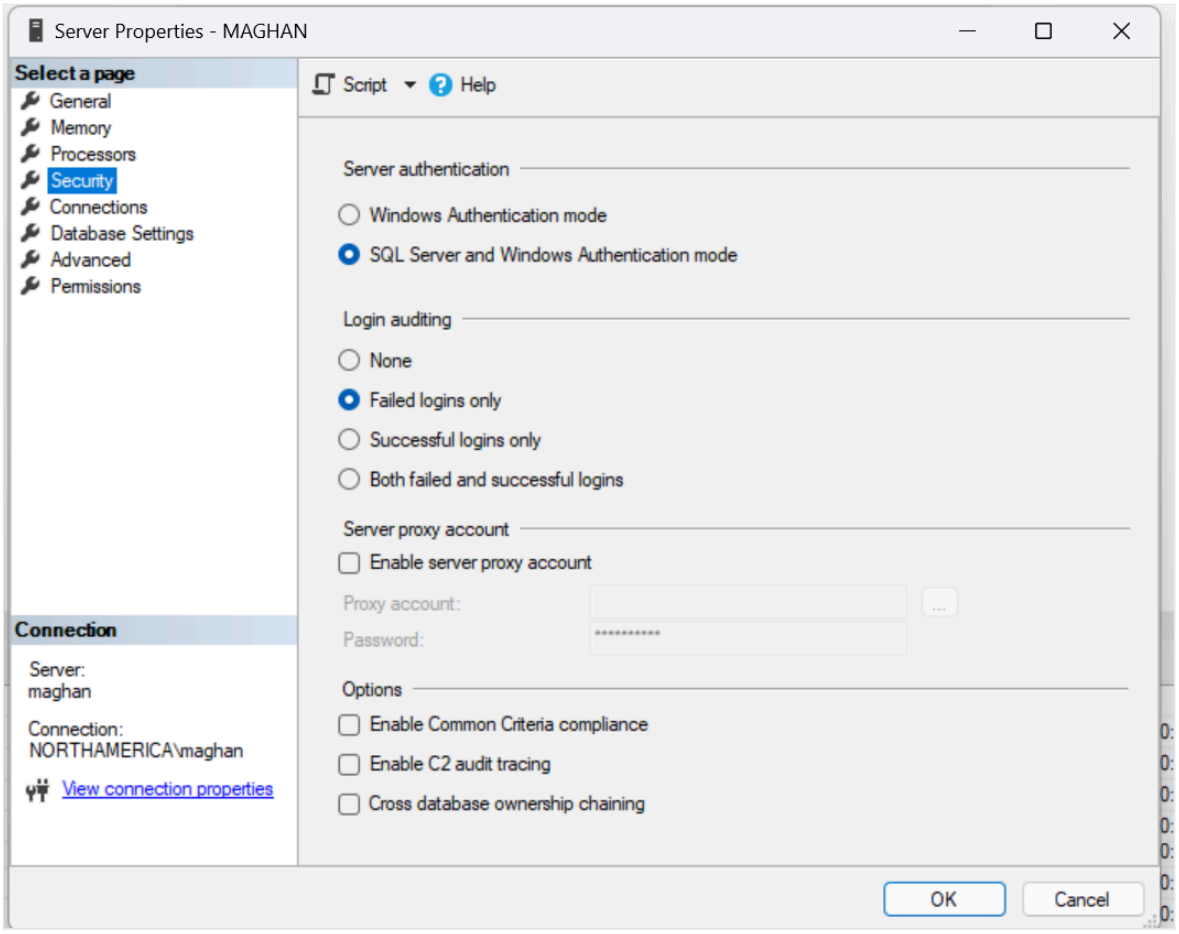
- Failed logins
- Successful logins
- Both failed and successful logins

You must restart SQL Server before this option will take effect.

Use SQL Server Management Studio

Configure login auditing

1. In SQL Server Management Studio, connect to an instance of the SQL Server Database Engine with Object Explorer.
2. In Object Explorer, right-click the server name, and then select **Properties**.
3. On the **Security** page, under **Login** auditing, select the desired option and close the **Server Properties** page.
4. In Object Explorer, right-click the server name, and then select **Restart**.



Related content

- SQL Server Management Studio components and configuration
- Configure SQL Server Agent
- Create a server audit and database audit specification

Feedback

Was this page helpful? Yes No

Workaround to enable copying from find and replace window

Article • 02/13/2025

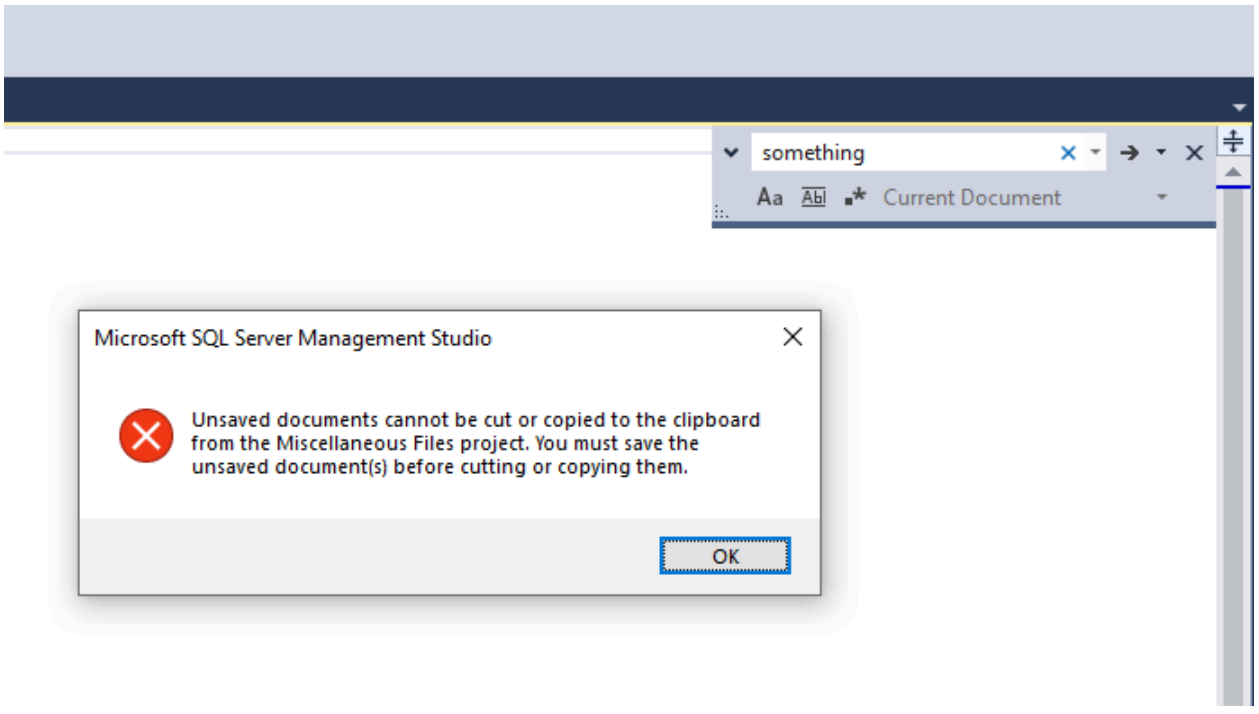
Applies to: SQL Server Azure SQL Database Azure Synapse Analytics Analytics Platform System (PDW)

A workaround is required to enable copying from find and replace window. If you find you are unable to copy from find and replace window in SQL Server Management Studio, follow the workaround below.

Error Message

When you attempt to copy text from the find and replace window in SQL Server Management Studio, you get an error message.

> Unsaved documents cannot be cut or copied to the clipboard from the miscellaneous files project. You must save the unsaved document(s) before cutting or copying them.



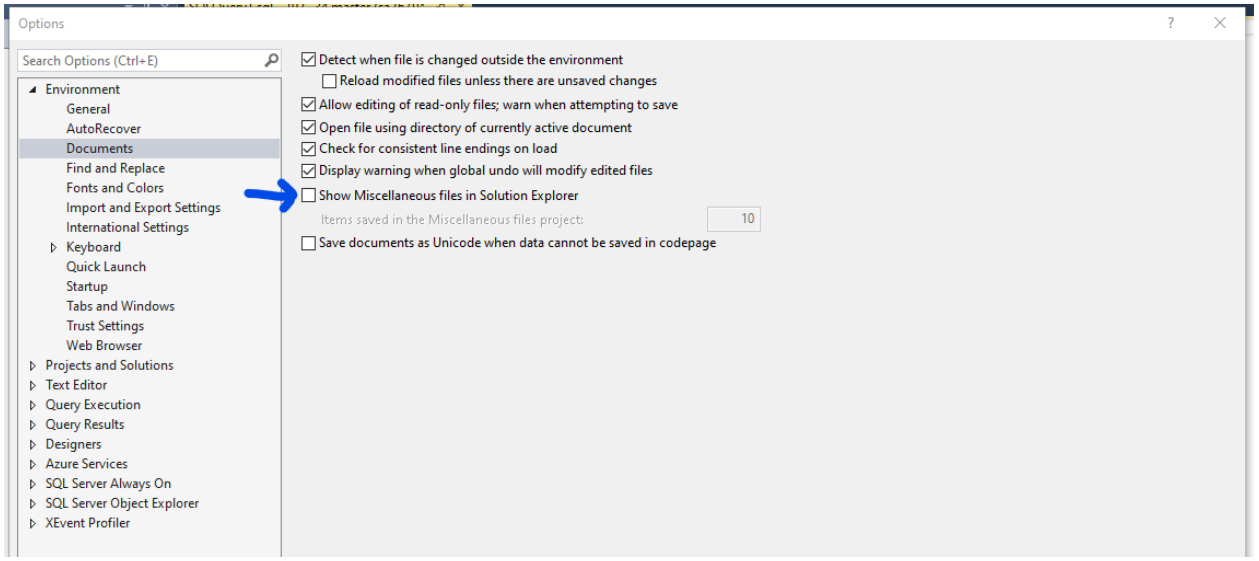
Workaround

To enable copying text from the find and replace window, follow these steps:

1. In the **Tools** menu, open **Options**.

2. Under **Environment**>**Documents**, uncheck the item for "Show Miscellaneous files in Solution Explorer"

3. Close and reopen SQL Server Management Studio



Feedback

Was this page helpful? Yes No

Workaround to move tabs

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure Synapse Analytics Analytics Platform System (PDW)

A workaround may be required to enable moving query editor tabs, whether within the window or to dock a previously removed tab. If you have applied all available Windows updates and find you are unable to manipulate query editor tabs without experiencing a crash, follow the workaround below.

Applicable Environments

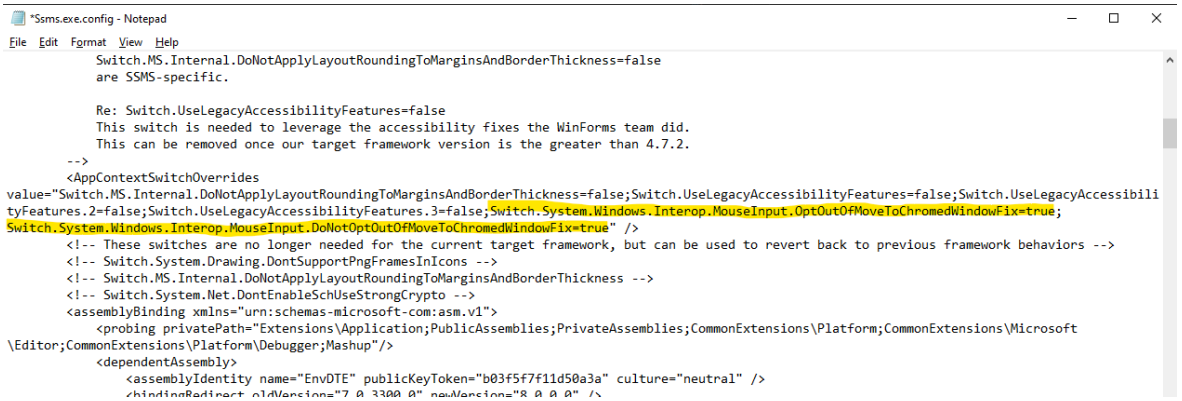
Windows updates to the .NET Framework introduced a known issue, which results in an application crash for SQL Server Management Studio (SSMS) when docking tabs or splitting the window. The latest information can be found at SQL Server User Feedback.

Workaround

If the crash persists after applying all available Windows updates, follow these steps to mitigate the issue:

1. Close all SQL Server Management Studio (SSMS) instances.
2. Locate your SSMS application file (exe). This is commonly found in `C:\Program Files (x86)\Microsoft SQL Server Management Studio 18\Common7\IDE`.
3. Open the file `Ssms.exe.config` in Notepad as Administrator.
4. Locate the `AppContextSwitchOverrides` node and append these two properties to the value.

```
;Switch.System.Windows.Interop.MouseInput.OptOutOfMoveToChromedWindowFix=true;
Switch.System.Windows.Interop.MouseInput.DoNotOptOutOfMoveToChromedWindowFix=true
```



5. Save the config file and reopen SSMS.

Feedback

Was this page helpful? Yes No

Configure link with SSMS - Azure SQL Managed Instance

Article • 10/09/2024

Applies to: Azure SQL Managed Instance

This article teaches you how to configure a link between SQL Server and Azure SQL Managed Instance by using SQL Server Management Studio (SSMS). With the link, databases from your initial primary are replicated to your secondary replica in near-real time.

After the link is created, you can then fail over to your secondary replica for the purpose of migration, or disaster recovery.

> **Note**
>
> - It's also possible to configure the link by using scripts.
> - Configuring Azure SQL Managed Instance as your initial primary is supported starting with SQL Server 2022 CU10.

Overview

Use the link feature to replicate databases from your initial primary to your secondary replica. For SQL Server 2022, the initial primary can be either SQL Server or Azure SQL Managed Instance. For SQL Server 2019 and earlier versions, the initial primary must be SQL Server. After the link is configured, the database from the initial primary is replicated to the secondary replica.

You can choose to leave the link in place for continuous data replication in a hybrid environment between the primary and secondary replica, or you can fail over the database to the secondary replica, to migrate to Azure, or for disaster recovery. For SQL Server 2019 and earlier versions, failing over to Azure SQL Managed Instance breaks the link and fail back is unsupported. With SQL Server 2022, you have the option to maintain the link and fail back and forth between the two replicas.

If you plan to use your secondary managed instance for only disaster recovery, you can save on licensing costs by activating the hybrid failover benefit.

Use the instructions in this article to manually set up the link between SQL Server and Azure SQL Managed Instance. After the link is created, your source database gets a read-only copy on your target secondary replica.

Prerequisites

To replicate your databases to your secondary replica through the link, you need the following prerequisites:

- An active Azure subscription. If you don't have one, create a free account.
- Supported version of SQL Server with required service update installed.
- Azure SQL Managed Instance. Get started if you don't have it.
- SQL Server Management Studio v19.2 or later.
- A properly prepared environment.

Consider the following:

- The link feature supports one database per link. To replicate multiple databases from an instance, create a link for each individual database. For example, to replicate 10 databases to SQL Managed Instance, create 10 individual links.
- Collation between SQL Server and SQL Managed Instance should be the same. A mismatch in collation could cause a mismatch in server name casing and prevent a successful connection from SQL Server to SQL Managed Instance.
- Error 1475 on your initial SQL Server primary indicates that you need to start a new backup chain by creating a full backup without the `COPY ONLY` option.
- To establish a link, or fail over, from SQL Managed Instance to SQL Server 2022, your managed instance must be configured with the SQL Server 2022 update policy. Data replication and failover from SQL Managed Instance to SQL Server 2022 is not supported by instances configured with the **Always-up-to-date** update policy.
- While you can establish a link from SQL Server 2022 to a SQL managed instance configured with the Always-up-to-date update policy, after failover to SQL Managed Instance, you will no longer be able to replicate data or fail back to SQL Server 2022.

Permissions

For SQL Server, you should have **sysadmin** permissions.

For Azure SQL Managed Instance, you should be a member of the SQL Managed Instance Contributor, or have the following custom role permissions:

Expand table

Microsoft.Sql/ resource	Necessary permissions
Microsoft.Sql/managedInstances	/read, /write
Microsoft.Sql/managedInstances/hybridCertificate	/action

Microsoft.Sql/ resource	Necessary permissions
Microsoft.Sql/managedInstances/databases	/read, /delete, /write, /completeRestore/action, /readBackups/action, /restoreDetails/read
Microsoft.Sql/managedInstances/distributedAvailabilityGroups	/read, /write, /delete, /setRole/action
Microsoft.Sql/managedInstances/endpointCertificates	/read
Microsoft.Sql/managedInstances/hybridLink	/read, /write, /delete
Microsoft.Sql/managedInstances/serverTrustCertificates	/write, /delete, /read

Prepare databases

If SQL Server is your initial primary, you need to create a backup of your database. Since Azure SQL Managed Instance takes backups automatically, skip this step if SQL Managed Instance is your initial primary.

Use SSMS to back up your database on SQL Server. Follow these steps:

1. Connect to your SQL Server in SQL Server Management Studio (SSMS).
2. In **Object Explorer**, right-click the database, hover over **Tasks** and then choose **Back up**.
3. Choose **Full** for backup type.
4. Ensure the **Back up to** option has the backup path to a disk with sufficient free storage space available.
5. Select **OK** to complete the full backup.

For more information, see Create a Full Database Backup.

> **Note**
>
> The link supports replicating user databases only. Replication of system databases is not supported. To replicate instance-level objects (stored in `master` or `msdb`), script them out and run T-SQL scripts on the destination instance.

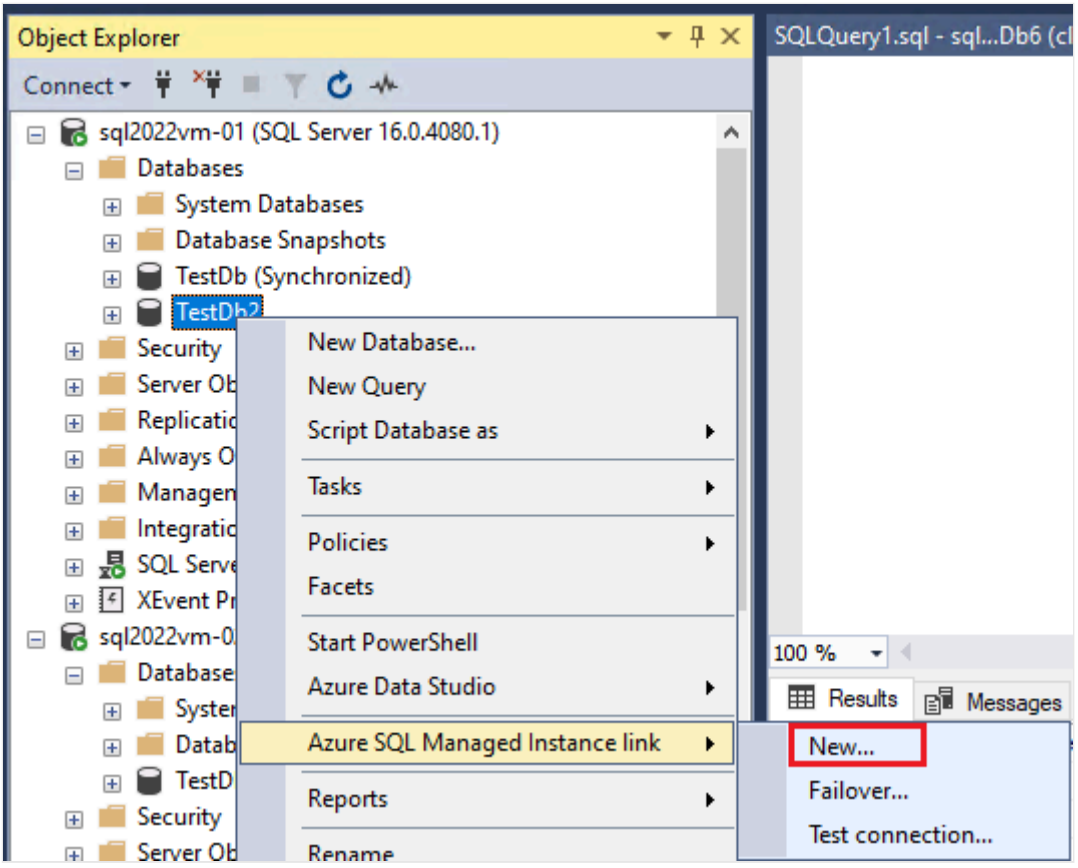
Create link to replicate database

In the following steps, use the **New Managed Instance link** wizard in SSMS to create a link between your initial primary and your secondary replica.

After you create the link, your source database gets a read-only copy on your target secondary replica.

1. Open SSMS and connect to your initial primary.

2. In **Object Explorer**, right-click the database you want to link to the secondary, hover over **Azure SQL Managed Instance link**, and select **New...** to open the **New Managed Instance link** wizard. If your SQL Server version isn't supported, the option isn't available in the context menu.



3. On the **Introduction** page of the wizard, select **Next**.

4. On the **Specify Link Options** page, provide a name for your link - if you select multiple databases, then the database name is automatically appended to the end of the name you provide so you don't have to include it yourself. Check the boxes if you want to enable connectivity troubleshooting and, for SQL Server 2022, if you plan to use the link for two-way disaster recovery. Select **Next**.

5. On the **Requirements** page, the wizard validates requirements to establish a link to your secondary. Select **Next** after all the requirements are validated, or resolve any requirements that aren't met and then select **Re-run Validation**.

6. On the **Select Databases** page, choose the database you want to replicate to your secondary replica via the link. Selecting multiple databases create multiple distributed availability groups, one for each link. Select **Next**.

7. On the **Specify Secondary Replica** page, select **Add secondary replica** add your secondary replica. If your initial primary is SQL Server, this opens the **Sign In** to Azure window. If your initial primary is SQL Managed Instance, this opens the **Connect to server** dialog box.
   a. For a SQL Server initial primary, sign in to Azure, choose the subscription, resource group, and secondary SQL Server managed instance from the dropdown. Select **Login** to open the **Connect to Server** dialog box and then connect to the SQL Managed Instance you want to replicate your database to. When you see **Login successful** on the **Sign in** window, select **OK** to close window and go back to the **New Managed Instance link** wizard.
   b. For a SQL Managed Instance initial primary, connect to the SQL Server instance you want to replicate your database to.

   > **Note**
   >
   > If you want establish a link to an availability group that already exists, then provide the IP address of the existing listener in the **Endpoint URL** field on the **Endpoints** tab of the **Specify Secondary Replica** page.

8. After adding your secondary replica, use the tabs in the wizard to modify **Endpoint** settings if you need to, and review information about backups and the link endpoint in the remaining tabs. Select **Next** when you're ready to proceed.

9. If SQL Managed Instance is your initial primary, the next page in the wizard is the **Login to Azure** page. Sign in again if you need to, and then select **Next**. This page isn't available when SQL Server is your initial primary.

10. On the **Validation** page, make sure all validations are successful. If any fail, resolve them, and then rerun the validation. Select **Next** when ready.

11. On the **Summary** page, review your configuration once more. Optionally, select **Script** to generate a script so you can easily recreate the same link in the future. Select **Finish** when you're ready to create the link.

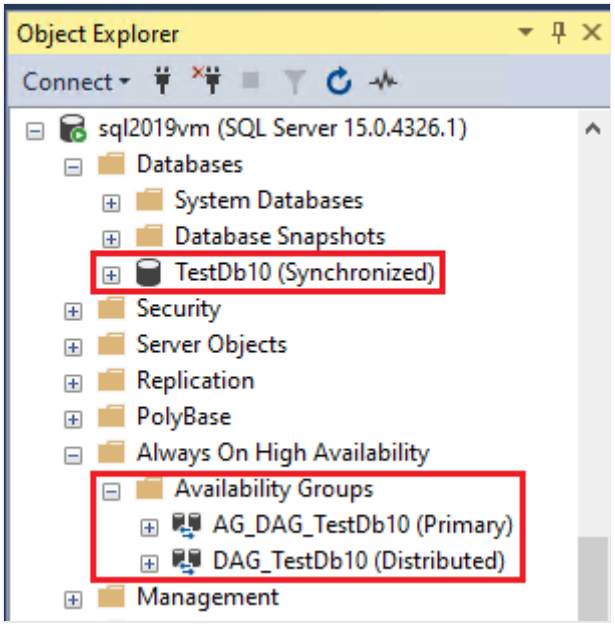
12. The **Executing actions** page displays the progress of each action.

13. After all steps finish, the **Results** page shows check marks next to the successfully completed actions. You can now close the window.

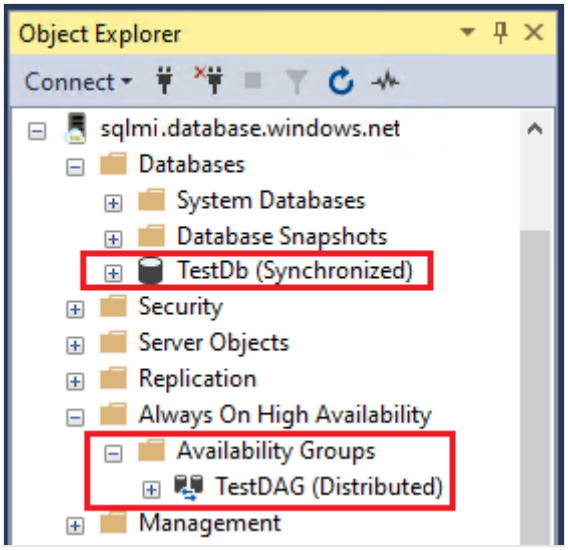
View a replicated database

After the link is created, your database is replicated to the secondary replica. Depending on database size and network speed, the database might initially be in a **Restoring** state on the secondary replica. After initial seeding finishes, the database is restored to the secondary replica and ready for read-only workloads.

On either replica, use **Object Explorer** in SSMS to view the **Synchronized** state of the replicated database.



Expand **Always On High Availability** and **Availability Groups** to view the distributed availability group created for each link.



Regardless of which instance is primary, you can also right-click the linked distributed availability group on SQL Server and select **Show Dashboard** to view the dashboard for the distributed availability group, which shows the status of the linked database in the distributed availability group.

Take first transaction log backup

If SQL Server is your initial primary, it's important to take the first transaction log backup on SQL Server *after* initial seeding completes, when the database is no longer in the **Restoring...** state on Azure SQL Managed Instance. Then take SQL Server transaction log backups regularly to minimize excessive log growth while SQL Server is in the primary role.

If SQL Managed Instance is your primary, you don't need to take any action as Azure SQL Managed Instance takes log backups automatically.

Drop a link

If you want to drop the link, either because it's no longer needed, or because it's in an irreparable state and needs to be recreated, you can do so with SQL Server Management Studio (SSMS).

You can delete the link from the following menu options in **Object Explorer** of SSMS, after connecting to your instance:

- **Always On Availability Groups** > **Availability Groups** > Right-click the distributed availability group name associated with the link > **Delete...**
- **Databases** > Right-click the database associated with the link > **Azure SQL Managed Instance link** > **Delete...**

Troubleshoot

If you encounter an error message when you create the link, select the error to open a window with additional details about the error.

If you encounter an error when working with the link, the SSMS wizard stops execution at the step that failed, and can't be restarted again. Address the issue, and, if necessary, clean up the environment to revert back to the original state by removing the distributed availability group and availability group if it was created while setting up the link. Then launch the wizard again to start over.

For more information, review troubleshoot issues with the link.

Related content

To use the link:

- Prepare environment for the Managed Instance link

- Configure link between SQL Server and SQL Managed instance with scripts
- Fail over the link
- Migrate with the link
- Best practices for maintaining the link
- Troubleshoot issues with the link

To learn more about the link:

- Managed Instance link overview
- Disaster recovery with Managed Instance link

For other replication and migration scenarios, consider:

- Transactional replication with SQL Managed Instance
- Log Replay Service (LRS)

SQL Server Management Studio (SSMS) setup failed or requires restarting the machine

Article • 02/13/2025

Infrequently, users encounter SSMS installation problems, and common error messages include:

- "Failed to install MSI package"
- "Microsoft ODBC Driver 17 for SQL Server: A previous installation required a reboot of the machine for changes to take effect"
- "Fatal error during installation (0x80070643)"

Suggested resolution

Following these steps to uninstall the "Microsoft ODBC Driver 17 for SQL Server" before beginning the SSMS installation commonly allows the setup to succeed if it fails with one of the aforementioned or similar error messages.

1. Close any related applications, including SSMS, Visual Studio, or SQL Server Profiler.
2. Go to Control Panel > Add/Remove Programs.
3. Locate the entry for "Microsoft ODBC Driver 17 for SQL Server" and uninstall. This step may require a restart.
4. Begin the SSMS installation. The latest version is available here.

Get help for SQL tools

- All the ways to get help
- SSMS user feedback .
- Submit an Azure Data Studio Git issue
- Contribute to Azure Data Studio
- SQL Client Tools Forum
- SQL Server Data Tools - MSDN forum
- Support options for business users

Related content

- Download SSMS

Feedback

Was this page helpful? Yes No

Get a full memory dump to troubleshoot SSMS

Article • 02/13/2025

Applies to: ✅ SQL Server ✅ Azure SQL Database ✅ Azure Synapse Analytics ⊗ Analytics Platform System (PDW)

In this article, you learn how to capture diagnostic information to troubleshoot a crash or an unresponsive system that you experience in SQL Server Management Studio (SSMS).

Get a full memory dump after an unresponsive system or crash

Get a full memory dump of SQL Server Management Studio (SSMS) when it stops responding or crashes.

To capture diagnostic information to troubleshoot a crash or an unresponsive SSMS, use the following steps:

1. Download ProcDump.
2. Unzip the download into a folder.
3. Open a Command Prompt (such as `cmd.exe`), and run the following command.

   ```
   Console

   <PathToProcDumpFolder>\procdump.exe -e -h -ma -w ssms.exe
   ```

   It should prompt you to accept a license agreement, select **Agree**.
4. Start SQL Server Management Studio (SSMS) if not started already.
5. Reproduce your issue.
6. Wait as the text appears in the cmd prompt about writing the dump file, don't proceed until it finishes.
7. Create a new folder and copy the *.dmp file that is written out to that folder.
8. Copy the following files into the same folder.

- "C:\Windows\Microsoft.NET\Framework\v4.0.30319\mscordacwks.dll"
- "C:\Windows\Microsoft.NET\Framework\v4.0.30319\SOS.dll"
- "C:\Windows\Microsoft.NET\Framework\v4.0.30319\clr.dll"

9. Zip up the folder.

OutOfMemoryException

You can also get the Full Memory Dump of SSMS when it throws an OutOfMemoryException (can be any managed exception).

To capture diagnostic information to troubleshoot an OutOfMemoryException from SSMS, use the following steps:

1. Download ProcDump.

2. Unzip the download into a folder.

3. Open Command Prompt and run the following command.

   Windows Command Prompt

   ```
   <PathToProcDumpFolder>\procdump.exe -e 1 -f System.OutOfMemoryException
   -ma -w ssms.exe
   ```

   It should prompt you to accept a license agreement, select **Agree**.

4. Start SQL Server Management Studio if not started already.

5. Repro the issue.

6. Wait as the text appears in the cmd prompt about writing the dump file, don't proceed until it finishes.

7. Create a new folder and copy the *.dmp file that is written out to that folder.

8. Copy the following files into the same folder.

   - "C:\Windows\Microsoft.NET\Framework\v4.0.30319\mscordacwks.dll"
   - "C:\Windows\Microsoft.NET\Framework\v4.0.30319\SOS.dll"
   - "C:\Windows\Microsoft.NET\Framework\v4.0.30319\clr.dll"

9. Zip up the folder.

Share the information

1. To share the information with the SSMS team, log the issue at the SSMS user feedback site.
2. Then share the memory dump file collected to OneDrive (or equivalent) where the file can be collected.

> **Important**
>
> Memory dump files may contain sensitive information.

Next steps

SQL Server Management Studio

Feedback

Was this page helpful? Yes No

Advanced troubleshooting for SQL Server Management Studio (SSMS)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure Synapse Analytics Analytics Platform System (PDW)

There are scenarios where trying to capture a memory dump for SSMS doesn't generate the expected output, and it requires advanced troubleshooting.

The following steps require Visual Studio (Community Edition or higher) to be installed.

To capture a diagnostic information with Visual Studio to troubleshoot a crash or an unresponsive SSMS, use the following steps:

1. Open Visual Studio.
2. Select **Continue without code** to open an empty window.
3. Start SSMS, if it's not already open.
4. Select **Debug** > **Attach to Process...**.
5. In the **Attach to Process** dialog, within the **Filter processes** box, enter SSMS.
6. In the list of processes select SSMS.exe and then **Attach**.
7. An Output window appears, with **Debug** selected for **Show output from:**.
8. Recreate the problematic behavior in SSMS.
9. Once SSMS closes, select **Debug** > **Save Dump As...** in Visual Studio and save the .dmp file to a folder.
10. Zip up the folder.
11. Stop debugging before closing Visual Studio.

Share the information

1. To share the information with the SSMS Team, log the issue at the SSMS user feedback site.

2. Then share the memory dump file collected to OneDrive (or equivalent) where the file can be collected.

> **Important**
>
> Memory dump files may contain sensitive information.

Enable verbose logging

The information logged from SSMS doesn't always provide enough detail for troubleshooting, and verbose logging can be enabled to capture more details.

1. Determine the location of the SSMS executable (ssms.exe). The default location for SSMS 20 is C:\Program Files (x86)\Microsoft SQL Server Management Studio 20\Common7\IDE, but may be different on your machine.

2. Open a command prompt and run the following, using the ssms.exe location in the previous step for the second line.

   Windows Command Prompt

   ```
   SET VsLogActivity=1
   "C:\Program Files (x86)\Microsoft SQL Server Management Studio
   20\Common7\IDE\ssms.exe"
   ```

3. SSMS starts.

4. Open Windows Explorer and navigate to %USERPROFILE%\AppData\Roaming\Microsoft\AppEnv\15.0.

5. Close SSMS to stop the verbose logging.

6. Inspect the ActivityLog.xml file which now contains more details that can assist with troubleshooting.

Clear SSMS cache files

Data stored in cache files may unexpectedly interfere with SSMS behavior. To rule out this problem, you can clear the files manually.

1. Close all instances of SSMS.

2. Remove all files in the following folders (it's recommended to make a copy of the RegSrvr*.xml file if you want to retain any entries under **Local Server Groups** in **Registered Servers**).

   - "%USERPROFILE%\AppData\Local\Microsoft\SQL Server Management Studio"
   - "%USERPROFILE%\AppData\Roaming\Microsoft\SQL Server Management Studio"

3. Start SSMS and observe if removing the cache files resolved the issue.

Related content

SQL Server Management Studio

Feedback

Was this page helpful? Yes No

Configure SQL Server Agent

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. Enabling and disabling SQL Server Agent is currently not supported in SQL Managed Instance. SQL Agent is always running. See SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to specify some configuration options for SQL Server Agent during installation of SQL Server. The full set of SQL Server Agent configuration options is only available within SQL Server Management Studio, SQL Server Management Objects (SMO), or the SQL Server Agent stored procedures.

Before you begin

Limitations and restrictions

- Select **SQL Server Agent** in Object Explorer of SQL Server Management Studio to administer jobs, operators, alerts, and the SQL Server Agent service. However, Object Explorer only displays the SQL Server Agent node if you have permission to use it.
- Auto-restart should not be enabled for the SQL Server service or the SQL Server Agent service on failover cluster instances.

Security

Permissions

To perform its functions, SQL Server Agent must be configured to use the credentials of an account that is a member of the **sysadmin** fixed server role in SQL Server. The account must have the following Windows permissions:

- Log on as a service (SeServiceLogonRight)

- Replace a process-level token (SeAssignPrimaryTokenPrivilege)
- Bypass traverse checking (SeChangeNotifyPrivilege)
- Adjust memory quotas for a process (SeIncreaseQuotaPrivilege)

For more information about the Windows permissions required for the SQL Server Agent service account, see Select an Account for the SQL Server Agent Service and Setting Up Windows Service Accounts.

To configure SQL Server Agent

1. Select the **Start** button, and then, on the **Start** menu, select **Control Panel**.
2. In Control Panel, select **System and Security**, select **Administrative Tools**, and then select **Local Security Policy**.
3. In Local Security Policy, select the chevron to expand the **Local Policies** folder, and then Select the **User Rights Assignment** folder.
4. Right-click a permission that you want to configure for use with SQL Server and select **Properties**.
5. In the permission's properties dialog box, verify that the account under which SQL Server Agent runs is listed. If not, select **Add User or Group**, enter the account under which SQL Server Agent runs in the **Select Users, Computers, Service Accounts, or Groups** dialog box, and then Select **OK**.
6. Repeat for each permission that you want to add to run with SQL Server Agent. When finished, select **OK**.

Feedback

Was this page helpful? Yes No

Assign a Job to a Job Category

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to assign Microsoft SQL Server Agent jobs to job categories in SQL Server by using SQL Server Management Studio, Transact-SQL or SQL Server Management Objects.

Job categories help you organize your jobs for easy filtering and grouping. For example, you can organize all your database backup jobs in the Database Maintenance category. You can assign jobs to built-in job categories, or you can create a user-defined job category and then assign jobs to that.

Before You Begin

Security

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To assign a job to a job category

1. In **Object Explorer**, click the plus sign to expand the server where you want to assign a job to a job category.
2. Click the plus sign to expand **SQL Server Agent**.
3. Click the plus sign to expand the **Jobs** folder.
4. Right-click the job you want to edit and select **Properties**.

5. In the **Job Properties** *-job_name* dialog box, in the **Category** list, select the job category you want to assign to the job.

6. Click **OK**.

Using Transact-SQL

To assign a job to a job category

1. In **Object Explorer**, connect to an instance of Database Engine.

2. On the Standard bar, click **New Query**.

3. Copy and paste the following example into the query window and click **Execute**.

```SQL
-- adding a new job category to the "NightlyBackups" job
USE msdb ;
GO
EXEC dbo.sp_update_job
    @job_name = N'NightlyBackups',
    @category_name = N'[Uncategorized (Local)]';
GO
```

For more information, see sp_update_job (Transact-SQL).

Using SQL Server Management Objects

To assign a job to a job category

Use the **JobCategory** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell.

Feedback

Was this page helpful? Yes No

Assign Alerts to an Operator

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to assign Microsoft SQL Server Agent alerts to operators so they can receive notifications about jobs in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Limitations and Restrictions

- SQL Server Management Studio provides an easy, graphical way to manage the entire alerting system. Using Management Studio is the recommended way to configure your alert infrastructure.
- To send a notification in response to an alert, you must first configure SQL Server Agent to send mail. For more information, see Configure SQL Server Agent Mail to Use Database Mail.
- If a failure occurs when sending an e-mail message or pager notification, the failure is reported in the SQL Server Agent service error log.

Security

Permissions

Only members of the **sysadmin** fixed server role can assign alerts to operators.

Using SQL Server Management Studio

To assign alerts to an operator

1. In **Object Explorer**, click the plus sign to expand the server that contains the operator to which you want to assign an alert.
2. Click the plus sign to expand **SQL Server Agent**.
3. Click the plus sign to expand the **Operators** folder.
4. Right-click the operator to which you want to assign an alert and select **Properties**, and select the **Notifications** page.
5. In the *operator_name* **Properties** dialog box, under **Select a page**, select **Notifications**.
6. Under **View notifications sent to this user by**, select **Alerts** to view a list of alerts sent to this operator or select **Jobs** to view a list of jobs that send notifications to this operator. Select one or more of the following checkboxes to define the notification method for each notification as necessary: **E-mail**, **Pager**, or **Net send**.
7. When finished, click **OK**.

Using Transact-SQL

To assign alerts to an operator

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- adds an e-mail notification for the specified alert (Test Alert)
-- This example assumes that Test Alert already exists
-- and that François Ajenstat is a valid operator name.
USE msdb ;
GO

EXEC dbo.sp_add_notification
 @alert_name = N'Test Alert',
 @operator_name = N'François Ajenstat',
 @notification_method = 1 ;
GO
```

For more information, see sp_add_notification (Transact-SQL).

Feedback

Was this page helpful? Yes No

Automatically Delete a Job

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to configure Microsoft SQL Server Agent in SQL Server to automatically delete jobs when they succeed, fail, or complete by using SQL Server Management Studio or SQL Server Management Objects.

Job responses ensure that database administrators know when jobs complete and how frequently they run. Typical job responses include:

- Notifying the operator by using e-mail, electronic paging, or a **net send** message.

  Use one of these job responses if the operator must perform a follow-up action. For example, if a backup job completes successfully, the operator must be notified to remove the backup tape and store it in a safe location.

- Writing an event message to the Windows application log.

  You can use this response only for failed jobs.

- Automatically deleting the job.

  Use this job response if you are certain that you do not need to rerun this job.

Before You Begin

Security

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To automatically delete a job

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**, expand **Jobs**, right-click the job you want to edit, and then click **Properties**.
3. Select the **Notifications** page.
4. Check **Automatically delete job**, and choose one of the following:
    - Click **When the job succeeds** to delete the job status when it has completed successfully.
    - Click **When the job fails** to delete the job when it has completed unsuccessfully.
    - Click **When the job completes** to delete the job regardless of completion status.

Using SQL Server Management Objects

To automatically delete a job

Use the **DeleteLevel** property of the **Job** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell. For more information, see SQL Server Management Objects (SMO).

Feedback

Was this page helpful? Yes No

Auto Restart SQL Server Agent

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to configure Microsoft SQL Server Agent to automatically restart if it should stop unexpectedly.

Before You Begin

Limitations and Restrictions

Object Explorer only displays the SQL Server Agent node if you have permission to use it.

Security

Permissions

To perform its functions, SQL Server Agent must be configured to use the credentials of an account that is a member of the **sysadmin** fixed server role in SQL Server. The account must have the following Windows permissions:

- Log on as a service (SeServiceLogonRight)
- Replace a process-level token (SeAssignPrimaryTokenPrivilege)
- Bypass traverse checking (SeChangeNotifyPrivilege)
- Adjust memory quotas for a process (SeIncreaseQuotaPrivilege)

> **Note**

For Auto Restart to function properly, the account running the SQL Server Agent service must be a local administrator on the server.

For more information about the Windows permissions required for the SQL Server Agent service account, see Select an Account for the SQL Server Agent Service and Setting Up Windows Service Accounts.

Using SQL Server Management Studio

To configure SQL Server Agent to automatically restart

1. In **Object Explorer**, click the plus sign to expand the server where you want to configure SQL Server Agent to automatically restart.
2. Right-click **SQL Server Agent**, and then click **Properties**.
3. On the **General** page, check **Auto restart SQL Server Agent if it stops unexpectedly**.

Feedback

Was this page helpful? Yes No

Change an Operator's Availability

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to change an operator's schedule for receiving alert notifications in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Security

Permissions

Only members of the **sysadmin** fixed server role can edit operators.

Using SQL Server Management Studio

To change an operator's availability

1. In **Object Explorer**, click the plus sign to expand server that contains the operator that you want to enable or disable.
2. Click the plus sign to expand **SQL Server Agent**.
3. Click the plus sign to expand the **Operators** folder.
4. Right-click the operator that you want to enable or disable and select **Properties**, then click the **General** tab.
5. In the *operator_name***Properties** dialog box, select or clear the **Enabled** check box.
6. Click **OK**.

Using Transact-SQL

To change an operator's availability

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- disables the 'François Ajenstat' operator
USE msdb ;
GO

EXEC dbo.sp_update_operator
    @name = N'François Ajenstat',
    @enabled = 0;
GO
```

For more information, see sp_update_operator (Transact-SQL).

Feedback

Was this page helpful? Yes No

Change the Membership of a Job Category

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to change the membership of the job category in SQL Server by using SQL Server Management Studio, Transact-SQL, or SQL Server Management Objects.

Job categories help you organize your jobs for easy filtering and grouping. You can create your own job categories. You can also change Microsoft SQL Server Agent jobs membership in job categories.

Before You Begin

Security

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To change the membership of a job category

1. In **Object Explorer**, click the plus sign to expand the server where you want to edit a job category.
2. Click the plus sign to expand **SQL Server Agent**.
3. Right-click the **Jobs** folder and select **Manage Job Categories**.
4. In the **Manage Job Categories***server_name* dialog box, select the job category that you want to edit, and then click **View Jobs**.

5. Select the **Show all jobs** check box.

6. To add a job to the category, in the main grid, select the check box in the **Select** column corresponding to the job. To remove a job from the category, clear the box. When finished, click **OK**.

7. Close the **Manage Job Categories***server_name* dialog box.

Using Transact-SQL

To change the membership of a job category

1. In **Object Explorer**, connect to an instance of Database Engine.

2. On the Standard bar, click **New Query**.

3. Copy and paste the following example into the query window and click **Execute**.

```
-- adding a new job category to the "NightlyBackups" job
USE msdb ;
GO
EXEC dbo.sp_update_job
    @job_name = N'NightlyBackups',
    @category_name = N'[Uncategorized (Local)]';
GO
```

For more information, see sp_update_job (Transact-SQL).

Using SQL Server Management Objects

To change the membership of a job category

Use the **JobCategory** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell.

Feedback

Was this page helpful? Yes No

Choose the Right SQL Server Agent Service Account for Multiserver Environments

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

The Windows account you choose for the SQL Server Agent service can affect the behavior of a multiserver environment, as follows:

- If you run the SQL Server Agent service under an account that is not a member of the local Windows Administrators group, enlisting target servers to master servers may fail. If it does, the following error message is returned:

  "The enlistment operation failed."

  Restart the SQL Server and the SQL Server Agent services to resolve this issue.

- When the SQL Server Agent service is run under the Local System account, master server-target server operations are supported only if both the master server and the target server reside on the same computer. If you use this configuration, the following message is returned when you enlist target servers to a master server:

  "Ensure the agent start-up account for *<target_server_computer_name>* has rights to log on as targetServer."

  You can ignore this informational message. The enlistment operation should complete successfully.

For more information about choosing an account for the SQL Server Agent service, see Select an Account for the SQL Server Agent Service.

Feedback

Was this page helpful? Yes No

Clear the Job History Log

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to delete the contents of the Microsoft SQL Server Agent job history log in SQL Server by using SQL Server Management Studio, Transact-SQL, or SQL Server Management Objects.

Before You Begin

Security

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To clear the job history log

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**, and then expand **Jobs**.
3. Right-click a job and click **View history**.
4. In the **Log File Viewer**, select the job for which you want to clear history, and then do one of the following:
    - Click **Delete**, and then click **Delete all history** in the **Delete History** dialog. You can delete all job history or only history that is older than a specified date. If you want to remove all job history, click **Delete all history**. If you only

want to remove older job history logs, click **Delete history before**, and then specify a date.

- Click **Job status** if you want to clear the history log of a multiserver job. Click **Job**, click a job name, and then click **View Remote Job History**.

5. Click **Delete**.

Using Transact-SQL

To clear the job history log

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- example removes the history for a job named NightlyBackups.
USE msdb ;
GO

EXEC dbo.sp_purge_jobhistory
    @job_name = N'NightlyBackups' ;
GO
```

Using SQL Server Management Objects

To clear the job history log

Use the **PurgeJobHistory** method of the **JobServer** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell. For more information, see SQL Server Management Objects (SMO).

Feedback

Was this page helpful? Yes No

Configure a User to Create and Manage SQL Server Agent Jobs

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to configure a user to create or execute Microsoft SQL Server Agent jobs.

- **Before you begin:** Security
- **To configure a user to create and manage SQL Server Agent jobs, using:** SQL Server Management Studio

Before You Begin

Security

To configure a user to create or execute Microsoft SQL Server Agent jobs, you must first add an existing SQL Server login or msdb role to one of the following SQL Server Agent fixed database roles in the msdb database: SQLAgentUserRole, SQLAgentReaderRole, or SQLAgentOperatorRole.

By default, members of these database roles can create their own job steps that run as themselves. If these non-administrative users want to run jobs that execute other job step types (for example, SSIS packages), they will need to have access to a proxy account. All members of the sysadmin fixed server role have permission to create, modify, and delete proxy accounts. For more information about the permissions that are associated with these SQL Server Agent fixed database roles, see SQL Server Agent Fixed Database Roles.

Permissions

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To add a SQL login or msdb role to a SQL Server Agent fixed database role

1. In **Object Explorer**, expand a server.
2. Expand **Security**, and then expand **Logins**.
3. Right-click the login you wish to add to a SQL Server Agent fixed database role, and select **Properties**.
4. On the **User Mapping** page of the **Login Properties** dialog box, select the row containing **msdb**.
5. Under **Database role membership for: msdb**, check the appropriate SQL Server Agent fixed database role.

To configure a proxy account to create and manage SQL Server Agent job steps

1. In **Object Explorer**, expand a server.
2. Expand **SQL Server Agent**.
3. Right-click **Proxies** and select **New Proxy**.
4. On the **General** page of the **New Proxy Account** dialog, specify the proxy name, credential name, and description for the new proxy. Note that you must create a credential first before creating a SQL Server Agent proxy. For more information about creating a credential, see How to: Create a Credential and CREATE CREDENTIAL (Transact-SQL).
5. Check the appropriate subsystems for this proxy.
   a. Operating system (CmdExec)
   b. SQL Server Analysis Services Query
   c. SQL Server Analysis Services Command
   d. SQL Server Integration Services Package
   e. PowerShell
6. On the **Principals** page, add or remove logins or roles to grant or remove access to the proxy account.

See Also

- Implement SQL Server Agent Security

Feedback

Was this page helpful? Yes No

Enlist a Target Server to a Master Server

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to add target servers to a multiserver administration configuration. Run this procedure from the master server. in SQL Server by using SQL Server Management Studio, Transact-SQL, or SQL Server Management Objects (SMO).

For information about how the Windows account used for the SQL Server Agent service affects a multiserver environment, see Create a Multiserver Environment.

Full Transport Layer Security (TLS), previously known as Secure Sockets Layer (SSL), encryption and certificate validation is enabled for connections between master servers and target servers by default. For more information, see Set Encryption Options on Target Servers.

Using SQL Server Management Studio

To enlist a target server

1. In **Object Explorer**, expand a server that is configured as a master server.
2. Right-click **SQL Server Agent**, point to **Multi Server Administration**, and then click **Add Target Servers**.
3. Complete the Target Server Wizard, which guides you through the process.

Using Transact-SQL

To enlist a target server

1. Use the **sp_msx_enlist** stored procedure. For more information, see sp_msx_enlist (Transact-SQL)

See Also

Automated Administration Across an Enterprise

Feedback

Was this page helpful? Yes No

Force a Target Server to Poll the Master Server

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This article describes how to force a target server to poll the master server. The target server must be a registered server on the master server.

A job is a specified series of actions that SQL Server Agent performs. A multiserver job is a job that a master server runs on one or more target servers. Each target server can run one instance of the same job at the same time. Each target server periodically polls the master server, downloads a copy of any new jobs assigned to the target server, and then disconnects. The target server runs the job locally and then reconnects to the master server to upload the job outcome status.

> **Note**
>
> If the master server is inaccessible when the target server tries to upload job status, the job status is spooled until the master server can be accessed.

- **Before you begin:** Limitations and Restrictions, Security
- **To force a target server to poll the master server, using:** SQL Server Management Studio

Before You Begin

Limitations and Restrictions

The target server must be a registered server on the master server. You must run the instructions given in this topic from the master server.

Security

For detailed information, see Implement SQL Server Agent Security and Choose the Right SQL Server Agent Service Account for Multiserver Environments.

Use SQL Server Management Studio

To force a target server to poll the master server

1. In **Object Explorer**, expand the master server.
2. Right-click **SQL Server Agent**, point to **Multi Server Administration**, and then select **Manage Target Servers**.
3. Select a target server, and then select **Force Poll**.

Feedback

Was this page helpful? Yes No

Format Pager Addresses for Alerts

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to format pager addresses for Microsoft SQL Server Agent alerts in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Security

Permissions

By default, members of the **sysadmin** fixed server role can view information about an alert. Other users must be granted the **SQLAgentOperatorRole** fixed database role in the **msdb** database.

Using SQL Server Management Studio

To format pager addresses

1. In **Object Explorer**, click the plus sign to expand the server that contains the alert that you want to send to a pager.
2. Right-click **SQL Server Agent** and select **Properties**
3. Under **Select a page**, select **Alert System**.
4. In the **To line** and **CC line** boxes of the **Address formatting for pager e-mails** field, enter the pager address prefix or suffix. The operator's actual pager address is inserted when a notification is sent.

5. In the **Subject** box, enter the subject line prefix or suffix.
6. Select the **Include body of e-mail in notification page** check box to include the full e-mail message with the pager message (as opposed to the subject line only).
7. When finished, click **OK**.

Feedback

Was this page helpful? Yes No

Give Others Ownership of a Job

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to reassign ownership of Microsoft SQL Server Agent jobs to another user.

- **Before you begin:** Limitations and Restrictions, Security
- **To give others ownership of a job, using:**

  SQL Server Management Studio

  Transact-SQL

  SQL Server Management Objects

Before You Begin

Limitations and Restrictions

To create a job, a user must be a member of one of the SQL Server Agent fixed database roles or the **sysadmin** fixed server role. A job can be edited only by its owner or members of the **sysadmin** role. For more information about the SQL Server Agent fixed database roles, see SQL Server Agent Fixed Database Roles.

You must be a system administrator to change the owner of a job.

Assigning a job to another login does not guarantee that the new owner has sufficient permission to run the job successfully.

Security

For security reasons, only the job owner or a member of the **sysadmin** role can change the definition of the job. Only members of the **sysadmin** fixed server role can assign job ownership to other users, and they can run any job, regardless of the job owner.

> ⓘ **Note**
>
> If you change job ownership to a user who is not a member of the **sysadmin** fixed server role, and the job is executing job steps that require proxy accounts (for example, SSIS package execution), make sure that the user has access to that proxy account or else the job will fail.

Permissions

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To give others ownership of a job

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**, expand **Jobs**, right-click the job, and then click **Properties**.
3. In the **Owner** list, select a login. You must be a system administrator to change the owner of a job.

   Assigning a job to another login does not guarantee that the new owner has sufficient permission to run the job successfully.

Using Transact-SQL

To give others ownership of a job

1. In Object Explorer, connect to an instance of the Database Engine, and then expand that instance.
2. On the toolbar, click **New Query**.
3. In the query window, enter the following statements that use the sp_manage_jobs_by_login (Transact-SQL) system stored procedure. The following

example reassigns all jobs from `danw` to `françoisa`.

```
USE msdb ;
GO

EXEC dbo.sp_manage_jobs_by_login
    @action = N'REASSIGN',
    @current_owner_login_name = N'danw',
    @new_owner_login_name = N'françoisa' ;
GO
```

Using SQL Server Management Objects

To give others ownership of a job

1. Call the **Job** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell. For example code, see Scheduling Automatic Administrative Tasks in SQL Server Agent.

See Also

Implement Jobs
Create Jobs

Feedback

Was this page helpful? Yes No

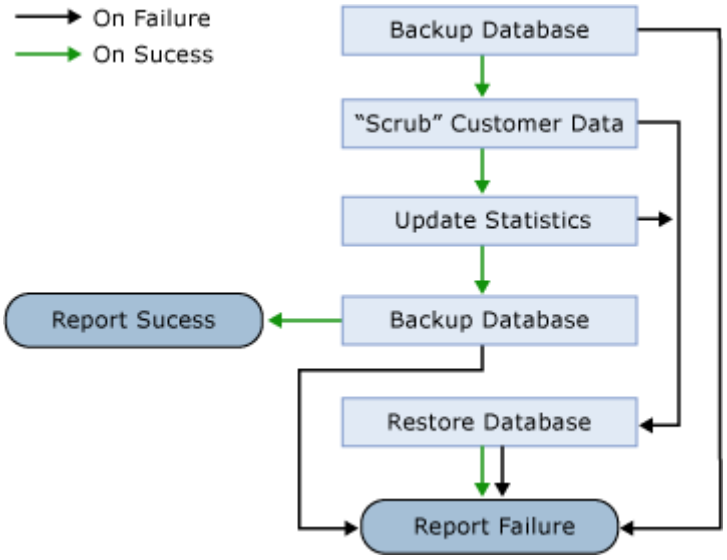
Handle Multiple Job Steps

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

If your job has more than one job step, you must specify the order in which the job steps run. This is called *control of flow*. You can add new job steps and rearrange the flow of job steps at any time; the changes take effect the next time the job is run. This illustration shows the control of flow for a database backup job.



The first step is Backup Database. If this step fails, SQL Server Agent reports failure to the operator who is defined to receive notification. If the Backup Database step succeeds, the job proceeds to the next step, "Scrub" Customer Data. If this step fails, SQL Server Agent skips forward to Restore Database. If "Scrub" Customer Data succeeds, the job proceeds to the next step, Update Statistics, and so on, until the final step either results in Report Success or Report Failure.

You define a control-of-flow action for the success and failure of each job step. You must specify an action to be taken when a job step succeeds and an action to be taken when a job step fails. You can also define the number of retry attempts for failed job steps and the interval between the retry attempts.

> **Note**
>
> When you use the SQL Server Agent graphical user interface (GUI) and delete one or more steps from a multistep job, the GUI removes all job steps and then adds the remaining steps back with the correct on-success or on-failure references. For example, suppose you have a job with five steps, and the first step is configured to jump to step 4 if it completes successfully. If you delete step 3, the GUI removes all steps for this job and adds the remaining four steps (1, 2, 4, and 5) with corrected references. In this case, the reference in step 1 would be reconfigured to jump to step 3 if step 1 completes successfully.

Job steps must be self-contained. That is, a job cannot pass Boolean values, data, or numeric values between job steps. You can, however, pass values from one Transact-SQL job step to another by using permanent tables or global temporary tables. You can pass values from job steps that run executable programs from one job step to another job step by using files. For example, the executable run by one job step writes a file, and the executable run by a subsequent job step reads the file.

> **Note**
>
> If you create looping job steps (job step 1 is followed by job step 2, then job step 2 returns to job step 1), a warning message appears when the job is created using SQL Server Management Studio.

SQL Server Agent records job and job step information in the job history.

See Also

sp_add_job
sysjobhistory
sysjobs (Transact-SQL)
sysjobsteps
Implement Jobs
Manage Job Steps

Feedback

Was this page helpful? Yes No

List Job Category Information

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to list job category information in SQL Server by using Transact-SQL or SQL Server Management Objects.

- **Before you begin:**

  Security

- **To list job category information, using:**

  Transact-SQL

  SQL Server Management Objects

Before You Begin

Security

For detailed information, see Implement SQL Server Agent Security.

Using Transact-SQL

To list job category information

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
```

```
-- returns information about jobs that are administered locally
USE msdb ;
GO

EXEC dbo.sp_help_category
    @type = N'LOCAL' ;
GO
```

For more information, see sp_help_category (Transact-SQL).

Using SQL Server Management Objects

To list job category information

Use the **JobCategory** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell.

Feedback

Was this page helpful? Yes No

Make a Master Server

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to make a master server SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Security

Distributed jobs that have steps which are associated with a proxy run under the context of the proxy account on the target server. Make sure that the following conditions are met or job steps that are associated with a proxy will not be downloaded from the master server to the target:

- The master server registry subkey **\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\<*instance_name*>\SQL Server Agent\AllowDownloadedJobsToMatchProxyName** (REG_DWORD) is set to 1 (true). By default, this subkey is set to 0 (false).

- A proxy account exists on the target server that has the same name as the master server proxy account under which the job step runs.

If job steps that use proxy accounts fail when downloading them from the master server to the target server, you can check the **error_message** column in the **sysdownloadlist** table in the **msdb** database for the following error messages:

- "The job step requires a proxy account, however proxy matching is disabled on the target server."

To resolve this error, set the **AllowDownloadedJobsToMatchProxyName** registry subkey to 1.

- "Proxy not found."

  To resolve this error, make sure a proxy account exists on the target server that has the same name as the master server proxy account under which the job step runs.

Permissions

Permissions to execute this procedure default to members of the **sysadmin** fixed server role.

Using SQL Server Management Studio

To make a master server

1. In **Object Explorer,** connect to an instance of the Microsoft SQL Server Database Engine, and then expand that instance.

2. Right-click **SQL Server Agent**, point to **Multi Server Administration**, and then click **Make this a Master**. The **Master Server Wizard** guides you through the process of making a master server and adding target servers.

3. From the **Master Server Operator** page, configure an operator for the master server To send notifications to operators by using e-mail or pagers, SQL Server Agent must be configured to send e-mail. To send notifications to operators by using **net send**, the Messenger service must be running on the server where SQL Server Agent resides.

   E-mail address
   Sets the e-mail address for the operator.

   Pager address
   Sets the pager e-mail address for the operator.

   Net send address
   Sets the **net send** address for the operator.

4. From the **Target Server** page, select target servers for the master server.

   Registered Servers
   Lists the servers registered in Microsoft SQL Server Management Studio that are

not already target servers.

Target Servers
Lists the servers that are target servers.

>
Move the selected server to the target server list.

>>
Move all servers to the target server list.

<
Remove the selected server from the target server list.

<<
Remove all servers from the target server list.

Add connection
Add a server to the target server list without registering the server.

Connection
Change the connection properties for the selected server.

5. From the **Master Server Login Credentials** page to specify if you want to create a new login for the target server, if necessary, and assign it rights to the master server.

   Create a new login if necessary and assign it rights to the MSX
   Create a new login on the target server if the login specified does not already exist.

Using Transact-SQL

To make a master server

1. Connect to the Database Engine.
2. From the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**. This example enlists the current server into the AdventureWorks1 master server. The location for the current server is Building 21, Room 309, Rack 5.

```
```

```
USE msdb ;
GO

EXEC dbo.sp_msx_enlist N'AdventureWorks1',
    N'Building 21, Room 309, Rack 5' ;
GO
```

For more information, see sp_msx_enlist (Transact-SQL).

See Also

- Create a Multiserver Environment
- Automated Administration Across an Enterprise

Feedback

Was this page helpful? Yes No

Make a Target Server

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to make a target server in SQL Server by using SQL Server Management Studio, Transact-SQL, or SQL Server Management Objects (SMO).

Before You Begin

Security

Distributed jobs that have steps which are associated with a proxy run under the context of the proxy account on the target server. Make sure that the following conditions are met or job steps that are associated with a proxy will not be downloaded from the master server to the target:

- The master server registry subkey **\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\<*instance_name*>\SQL Server Agent\AllowDownloadedJobsToMatchProxyName** (REG_DWORD) is set to 1 (true). By default, this subkey is set to 0 (false).

- A proxy account exists on the target server that has the same name as the master server proxy account under which the job step runs.

If job steps that use proxy accounts fail when downloading them from the master server to the target server, you can check the **error_message** column in the **sysdownloadlist** table in the **msdb** database for the following error messages:

- "The job step requires a proxy account, however proxy matching is disabled on the target server."

To resolve this error, set the **AllowDownloadedJobsToMatchProxyName** registry subkey to 1.

- "Proxy not found."

  To resolve this error, make sure a proxy account exists on the target server that has the same name as the master server proxy account under which the job step runs.

Permissions

Permissions to execute this procedure default to members of the **sysadmin** fixed server role.

Using SQL Server Management Studio

To make a target server

1. In **Object Explorer,** connect to an instance of the Microsoft SQL Server Database Engine, and then expand that instance.
2. Right-click **SQL Server Agent**, point to **Multi Server Administration**, and then click **Make this a Target**. The **Target Server Wizard** guides you through the process of making a target server.
3. From the **Select a Master Server** page, select the master server that this target server will receive jobs from.

   Pick Server
   Connect to the master server.

   Description of this server
   Type a description for this target server. The target server uploads this description to the master server.
4. From the **Master Server Login Credentials** page, create a new login on the target server, if necessary.

   Create a new login if necessary and assign it rights to the MSX
   Create a new login on the target server if the login specified does not already exist.

Using Transact-SQL

To make a target server

1. Connect to the Database Engine.

2. From the Standard bar, click **New Query**.

3. Copy and paste the following example into the query window and click **Execute**. This example enlists the current server into the AdventureWorks1 master server. The location for the current server is Building 21, Room 309, Rack 5.

   ```
   SQL
   ```

   ```
   USE msdb ;
   GO

   EXEC dbo.sp_msx_enlist N'AdventureWorks1',
       N'Building 21, Room 309, Rack 5' ;
   GO
   ```

   For more information, see sp_msx_enlist (Transact-SQL).

See Also

Automated Administration Across an Enterprise

Feedback

Was this page helpful? Yes No

Manage Events

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

You can forward to an instance of SQL Server all event messages that meet or exceed a specific error severity level. This is called *event forwarding*. The forwarding server is a dedicated server that can also be a master server. You can use event forwarding to centralize alert management for a group of servers, thereby reducing the workload on heavily used servers.

When one server receives events for a group of other servers, the server that receives events is called an *alerts management server*. In a multiserver environment, you designate the master server as the alerts management server.

Advantages of Using an Alerts Management Server

The advantages of setting up an alerts management server include:

- **Centralization**. Centralized control and a consolidated view of the events of several instances of SQL Server are possible from a single server.
- **Scalability**. Many physical servers can be administered as one logical server. You can add or remove servers to this physical server group as needed.
- **Efficiency**. Configuration time is reduced, because you need to define alerts and operators only once.

Disadvantages of Using an Alerts Management Server

The disadvantages of setting up an alerts management server include:

- **Increased traffic**. Forwarding events to an alerts management server can increase network traffic. This increase can be moderated by restricting event forwarding to events that are above a designated severity level.
- **Single point of failure**. If the alerts management server goes offline, no alerts are issued for any event on the managed group of servers.
- **Server load**. Handling alerts for the forwarded events causes an increased processing load on the alerts management server.

Guidelines for Using an Alerts Management Server

When configuring an alerts management server, follow these guidelines:

- In order to receive forwarded events, the alerts management server must be a default instance of SQL Server.
- Avoid running critical or heavily used applications on the alerts management server.
- Carefully plan for the network traffic involved in configuring many servers to share the same alerts management server. If congestion results, reduce the number of servers that use a particular alerts management server.

  The servers that are registered within SQL Server Management Studio constitute the list of servers available to be chosen by that server as the alerts-forwarding server.
- Define alerts on the local instance of SQL Server that require a server-specific response, instead of forwarding the alerts to the alerts management server.

  The alerts management server views all the servers forwarding to it as a logical whole. For example, an alerts management server responds in the same way to a 605 event from server A and a 605 event from server B.
- After configuring your alert system, periodically check the Microsoft Windows application log for SQL Server Agent events.

  Failure conditions encountered by the alerts engine are written to the local Windows application log with a source name of "SQL Server Agent." For example, if SQL Server Agent cannot send an e-mail notification as it has been defined, an event is logged in the application log.

If a locally defined alert is inactivated, and an event occurs that would have caused the alert to fire, the event is forwarded to the alerts management server (if it satisfies the alert-forwarding condition). This forwarding allows local overrides (alerts defined locally that are also defined on the alerts management server) to be turned off and on as needed by the user at the local site. You can also request that events always be forwarded, even if they are also handled by local alerts.

The following are common tasks for managing events in a multiserver environment:

To designate an alerts management server

- SQL Server Management Studio

To define the response to an alert

- SQL Server Management Studio
- Transact-SQL

Running Event-Triggered Jobs

You can define a job to be executed in response to an alert. For example, you can execute a job that corrects or further diagnoses a problem detected by the alert.

> **Note**
>
> Because a job can raise an event, be careful not to create a recursive alert-job loop.

See Also

sp_add_notification (Transact-SQL)

Feedback

Was this page helpful? Yes No

Manage Job Steps

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

A job step is an action that the job takes on a database or a server. Every job must have at least one job step. Job steps can be:

- Executable programs and operating system commands.
- Transact-SQL statements, including stored procedures and extended stored procedures.
- PowerShell scripts.
- Microsoft ActiveX scripts.
- Replication tasks.
- Analysis Services tasks.
- Integration Services packages.

Every job step runs in a specific security context. If the job step specifies a proxy, the job step runs in the security context of the credential for the proxy. If a job step does not specify a proxy, the job step runs in the context of the SQL Server Agent service account. Only members of the sysadmin fixed server role can create jobs that do not explicitly specify a proxy.

Because job steps run in the context of a specific Microsoft Windows user, that user must have the permissions and configuration necessary for the job step to execute. For example, if you create a job that requires a drive letter or a Universal Naming Convention (UNC) path, the job steps may run under your Windows user account while testing the tasks. However, the Windows user for the job step must also have the necessary permissions, drive letter configurations, or access to the required drive. Otherwise, the job step fails. To prevent this problem, ensure that the proxy for each job

step has the necessary permissions for the task that the job step performs. For more information, see Security and Protection (Database Engine).

Job Step Logs

SQL Server Agent can write output from some job steps either to an operating system file or to the sysjobstepslogs table in the msdb database. The following job step types can write output to both destinations:

- Executable programs and operating system commands.
- Transact-SQL statements.
- Analysis Services tasks.

Only job steps that are executed by users who are members of the sysadmin fixed server role can write job step output to operating system files. If job steps are executed by users who are members of the SQLAgentUserRole, SQLAgentReaderRole, or the SQLAgentOperatorRole fixed database roles in the msdb database, then the output from these job steps can be written only to the sysjobstepslogs table.

Job step logs are automatically deleted when jobs or job steps are deleted.

> **Note**
>
> Replication task and Integration Services package job step logging is handled by their respective subsystem. You cannot use SQL Server Agent to configure jog step logging for these types of job steps.

Executable Programs and Operating-System Commands As Job Steps

Executable programs and operating-system commands can be used as job steps. These files may have .bat, .cmd, .com, or .exe file extensions.

When you use an executable program or an operating-system command as a job step, you must specify:

- The process exit code returned if the command was successful.
- The command to execute. To execute an operating system command, this is simply the command itself. For an external program, this is the name of the program and

the arguments to the program, for example: **C:\Program Files\Microsoft SQL Server\100\Tools\Binn\sqlcmd.exe -e -q "sp_who"**

> ⓘ Note
>
> You must provide the full path to the executable if the executable is not located in a directory specified in the system path or the path for the user that the job step runs as.

Transact-SQL Job Steps

When you create a Transact-SQL job step, you must:

- Identify the database in which to run the job.
- Type the Transact-SQL statement to execute. The statement may call a stored procedure or an extended stored procedure.

Optionally, you can open an existing Transact-SQL file as the command for the job step.

Transact-SQL job steps do not use SQL Server Agent proxies. Instead, the job step runs as the owner of the job step, or as the SQL Server Agent service account if the owner of the job step is a member of the sysadmin fixed server role. Members of the sysadmin fixed server role can also specify that Transact-SQL job steps run under the context of another user by using the *database_user_name* parameter of the sp_add_jobstep stored procedure. For more information, see sp_add_jobstep (Transact-SQL).

> ⓘ Note
>
> A single Transact-SQL job step can contain multiple batches. Transact-SQL job steps can contain embedded GO commands.

PowerShell Scripting Job Steps

When you create a PowerShell script job step, you must specify one of two things as the command for the step:

- The text of a PowerShell script.
- An existing PowerShell script file to be opened.

The SQL Server Agent PowerShell subsystem opens a PowerShell session and loads the SQL Server PowerShell snap-ins. The PowerShell script used as the job step command can reference the SQL Server PowerShell provider and cmdlets. For more information about writing PowerShell scripts using the SQL Server PowerShell snap-ins, see the SQL Server PowerShell.

ActiveX Scripting Job Steps

> **ⓘ Important**
>
> The ActiveX scripting job step will be removed from SQL Server Agent in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature.

When you create an ActiveX scripting job step, you must:

- Identify the scripting language in which the job step is written.
- Write the ActiveX script.

You can also open an existing ActiveX script file as the command for the job step. Alternatively, ActiveX script commands can be externally compiled (for example, using Microsoft Visual Basic) and then run as executable programs.

When a job step command is an ActiveX script, you can use the SQLActiveScriptHost object to print output to the job step history log or create COM objects. SQLActiveScriptHost is a global object that is introduced by SQL Server Agent hosting system into the script name space. The object has two methods (Print and CreateObject). The following example shows how ActiveX scripting works in Visual Basic Scripting Edition (VBScript).

```
' VBScript example for ActiveX Scripting job step
' Create a Dmo.Server object. The object connects to the
' server on which the script is running.

Set oServer = CreateObject("SQLDmo.SqlServer")
oServer.LoginSecure = True
oServer.Connect "(local)"
'Disconnect and destroy the server object
oServer.DisConnect
Set oServer = nothing
```

Replication Job Steps

When you create publications and subscriptions using replication, replication jobs are created by default. The type of job created is determined by the type of replication (snapshot, transactional, or merge) and the options used.

Replication job steps activate one of these replication agents:

- Snapshot Agent (Snapshot job)
- Log Reader Agent (LogReader job)
- Distribution Agent (Distribution job)
- Merge Agent (Merge job)
- Queue Reader Agent (QueueReader job)

When replication is set up, you can specify to run the replication agents in one of three ways: continuously after SQL Server Agent is started, on demand, or according to a schedule. For more information about replication agents, see Replication Agents Overview.

Analysis Services Job Steps

SQL Server Agent supports two distinct types of Analysis Services job steps, command job steps, and query job steps.

Analysis Services Command Job Steps

When you create an Analysis Services command job step, you must:

- Identify the database OLAP server in which to run the job step.
- Type the statement to execute. The statement must be an XML for Analysis Services **Execute** method. The statement may not contain a complete SOAP envelope or an XML for Analysis Services **Discover** method. Notice that, while SQL Server Management Studio supports complete SOAP envelopes and the **Discover** method, SQL Server Agent job steps do not.

Analysis Services Query Job Steps

When you create an Analysis Services query job step, you must:

- Identify the database OLAP server in which to run the job step.
- Type the statement to execute. The statement must be a multidimensional expressions (MDX) query.

For more information on MDX, see MDX Statement Fundamentals (MDX).

Integration Services Packages

When you create an Integration Services package job step, you must do the following:

- Identify the source of the package.
- Identify the location of the package.
- If configuration files are required for the package, identify the configuration files.
- If command files are required for the package, identify the command files.
- Identify the verification to use for the package. For example, you can specify that the package must be signed, or that the package must have a specific package ID.
- Identify the data sources for the package.
- Identify the log providers for the package.
- Specify variables and values to set before running the package.
- Identify execution options.
- Add or modify command-line options.

Note that if you deployed the package to the SSIS Catalog and you specify **SSIS Catalog** as the package source, much of this configuration information is obtained automatically from the package. Under the **Configuration** tab you can specify the environment, parameter values, connection manager values, property overrides, and whether the package runs in a 32-bit runtime environment.

For more information about creating job steps that run Integration Services packages, see SQL Server Agent Jobs for Packages.

Related Tasks

Expand table

Description	Topic
Describes how to create a job step with an executable program.	Create a CmdExec Job Step
Describes how to reset SQL Server Agent permissions.	Configure a User to Create and Manage SQL Server Agent Jobs
Describes how to create a Transact-SQL job step.	Create a Transact-SQL Job Step
Describes how to define options for Microsoft SQL Server Agent Transact-SQL job steps.	Define Transact-SQL Job Step Options
Describes how to create an ActiveX script job step.	Create an ActiveX Script Job Step
Describes how to create and define SQL Server Agent job steps that execute SQL Server Analysis Services commands and queries.	Create an Analysis Services Job Step
Describes what action SQL Server should take if a failure occurs during job execution.	Set Job Step Success or Failure Flow
Describes how to view job step details in the Job Step Properties dialog.	View Job Step Information
Describes how to delete a SQL Server Agent job step log.	Delete a Job Step Log

See Also

sysjobstepslogs (Transact-SQL)
Create Jobs
sp_add_job (Transact-SQL)

Feedback

Was this page helpful? Yes No

Manage Jobs Across an Enterprise

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

If you make changes to multiserver job definitions outside Microsoft SQL Server Management Studio, you must post the changes to the download list so that target servers can download the updated job again. To ensure that target servers have current job definitions, post an INSERT instruction after you update the multiserver job, as follows:

```
EXECUTE sp_post_msx_operation 'INSERT', 'JOB', '<job id>'
```

To notify target servers that a multiserver job has been modified, you must invoke the preceding command after using any of the following procedures:

- sp_add_jobstep (Transact-SQL)
- sp_update_jobstep (Transact-SQL)
- sp_delete_jobstep (Transact-SQL)
- Managing Events
- sp_detach_schedule (Transact-SQL)

> **Note**
>
> It is not necessary to call **sp_post_msx_operation** after you call **sp_update_job** or **sp_delete_job**, because these stored procedures post the required changes to the download list automatically.

The following are common tasks for managing jobs across an enterprise:

To check the status of a target server

- Transact-SQL
- SQL Server Management Objects (SMO)

To change the target servers for a job

- SQL Server Management Studio
- Transact-SQL
- SQL Server Management Objects (SMO)

To change the location of a target server

- SQL Server Management Studio
- Transact-SQL
- SQL Server Management Objects (SMO)

To synchronize target server clocks

- SQL Server Management Studio
- Transact-SQL

To force a target server to poll the master server

- SQL Server Management Studio
- Transact-SQL

See Also

Manage Events

Feedback

Was this page helpful? Yes No

Manage Schedules

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Allows you to view and change properties for Microsoft SQL Server Agent job schedules.

Options

Available schedules
Lists the schedules available for this user. Notice that a job and a schedule must have the same owner. Therefore, this list only includes schedules owned by the owner of the job.

Name
Displays the name of the schedule.

Enabled
Select this option to enable the schedule.

Description
Describes the conditions under which the schedule runs the job.

Jobs in schedule
Lists the job numbers attached to the schedule. Click a number to view the properties of the job.

New
Click this button to create a new schedule.

Delete
Click this button to delete the selected schedule.

Properties
Click this button to change the properties of the selected schedule.

See Also

Create and Attach Schedules to Jobs

Feedback

Was this page helpful? Yes No

Modify a Job

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to change the properties of Microsoft SQL Server Agent jobs in SQL Server by using SQL Server Management Studio, Transact-SQL, or SQL Server Management Objects.

Before You Begin

Limitations and Restrictions

A SQL Server Agent master job cannot be targeted at both local and remote servers.

Security

Unless you are a member of the **sysadmin** fixed server role, you can only modify jobs that you own. For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To modify a job

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**, expand **Jobs**, right-click the job you want to modify, and then click **Properties**.
3. In the **Job Properties** dialog box, update the job's properties, steps, schedule, alerts, and notifications using the corresponding pages.

Using Transact-SQL

To modify a job

1. In Object Explorer, connect to an instance of the Database Engine, and then expand that instance.
2. On the toolbar, click **New Query**.
3. In the query window, use the following system stored procedures to modify a job.
    - Execute sp_update_job (Transact-SQL) to change the attributes of a job.
    - Execute sp_update_schedule (Transact-SQL) to change the scheduling details for a job definition.
    - Execute sp_add_jobstep (Transact-SQL) to add new job steps.
    - Execute sp_update_jobstep (Transact-SQL) to change pre-existing job steps.
    - Execute sp_delete_jobstep (Transact-SQL) to remove a job step from a job.
    - Additional stored procedures to modify any SQL Server Agent master job:
        - Execute sp_delete_jobserver (Transact-SQL) to delete a server currently associated with a job.
        - Execute sp_add_jobserver (Transact-SQL) to associate a server with the current job.

Using SQL Server Management Objects

To modify a job

Use the **Job** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell. For more information, see SQL Server Management Objects (SMO).

Feedback

Was this page helpful? Yes No

Modify a SQL Server Agent Master Job

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

The following topics describe how to modify a Microsoft SQL Server Agent master job.

Change the Scheduling Details for a SQL Server Agent Master Job

Add Steps to a SQL Server Agent Master Job

Change Steps of a SQL Server Agent Master Job

Remove Steps from a SQL Server Agent Master Job

Modify the Target Server(s) Associated with a SQL Server Agent Master Job

Feedback

Was this page helpful? Yes No

Modify a SQL Server Agent Proxy

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to modify a Microsoft SQL Server Agent proxy in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Limitations and Restrictions

- SQL Server Agent proxies use credentials to store information about Windows user accounts. The user specified in the credential must have "Log on as a batch job" permission on the computer on which SQL Server is running.
- SQL Server Agent checks subsystem access for a proxy and gives access to the proxy each time the job step runs. If the proxy no longer has access to the subsystem, the job step fails. Otherwise, SQL Server Agent impersonates the user that is specified in the proxy and runs the job step.
- If the login for the user has access to the proxy, or the user belongs to any role with access to the proxy, the user can use the proxy in a job step.

Security

Permissions

Only members of the **sysadmin** fixed server role can create, modify, or delete proxy accounts.

Using SQL Server Management Studio

To modify a SQL Server Agent proxy

1. In **Object Explorer**, click the plus sign to expand the server that contains the SQL Server Agent proxy account that you want to modify.
2. Click the plus sign to expand **SQL Server Agent**.
3. Click the plus sign to expand the **Proxies** folder.
4. Click the plus sign to expand the subsystem node for the proxy (for example, **ActiveX Script**).
5. Right-click the proxy account you want to modify and select **Properties**.
6. In the *proxy_name***Proxy Account Properties** dialog box, make changes to the proxy account as necessary. For more information on the options in this dialog box, see Create a SQL Server Agent Proxy.
7. When finished, click **OK**.

Using Transact-SQL

To modify a SQL Server Agent proxy

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
SQL
```

```
-- Disables the proxy named 'Catalog application proxy'.
USE msdb ;
GO
EXEC dbo.sp_update_proxy
    @proxy_name = 'Catalog application proxy',
    @enabled = 0;
GO
```

For more information, see sp_update_proxy (Transact-SQL).

Feedback

Was this page helpful? Yes No

Modify the Target Server(s) Associated with a SQL Server Agent Master Job

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to modify the target server(s) associated with a SQL Server Agent master job in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Limitations and Restrictions

A SQL Server Agent master job cannot be targeted at both local and remote servers.

Security

Permissions

Unless you are a member of the **sysadmin** fixed server role, you can only modify jobs that you own. For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To modify the target server(s) associated with a SQL Server Agent master job

1. In **Object Explorer,** click the plus sign to expand the server that contains the job where you want to modify the target server.

2. Click the plus sign to expand **SQL Server Agent**.
3. Click the plus sign to expand the **Jobs** folder.
4. Right-click the job where you want to modify the target server and select **Properties**.
5. In the **Job Properties** *-job_name* dialog box, under **Select a page**, select **Targets**. For more information on the available options on this page, see Job Properties - New Job (Targets Page).
6. When finished, click **OK**.

Using Transact-SQL

To delete a target server currently associated with a SQL Server Agent master job

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- removes the server SEATTLE2 from processing the Weekly Sales
Backupsjob
-- assumes that the Weekly Sales Backups job exists
USE msdb ;
GO

EXEC sp_delete_jobserver
    @job_name = N'Weekly Sales Backups',
    @server_name = N'SEATTLE2' ;
GO
```

For more information, see sp_delete_jobserver (Transact-SQL).

To associate a target server with the current SQL Server Agent master job

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.

3. Copy and paste the following example into the query window and click **Execute**.

```
-- assigns the multiserver job Weekly Sales Backups to the server
SEATTLE2
-- assumes that the Weekly Sales Backups job already exists and that
-- SEATTLE2 is registered as a target server for the current instance
USE msdb ;
GO

EXEC dbo.sp_add_jobserver
    @job_name = N'Weekly Sales Backups',
    @server_name = N'SEATTLE2' ;
GO
```

For more information, see sp_add_jobserver (Transact-SQL).

Feedback

Was this page helpful? Yes No

Modify the Target Servers for a Job

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to change the target servers for Microsoft SQL Server Agent jobs in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Security

Permissions

By default, members of the sysadmin fixed server role can execute this stored procedure. Other users must be granted one of the following SQL Server Agent fixed database roles in the msdb database:

1. **SQLAgentUserRole**
2. **SQLAgentReaderRole**
3. SQLAgentOperatorRole

Using SQL Server Management Studio

To modify the target servers for a job

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**, expand **Jobs**, right-click a job, and then click **Properties**.

3. In the **Job Properties** dialog, select the **Targets**page, and click **Target local server**, or **Target multiple servers**.

   If you choose **Target multiple servers**, designate the servers that will be targets for the job by checking the box to the left of the server name. Verify that the check boxes for servers that will not be targets of the job are unchecked.

Using Transact-SQL

To modify the target servers for a job

1. Connect to the Database Engine.
2. From the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**. This example assigns the multiserver job Weekly Sales Backups to the server SEATTLE2.

```
USE msdb ;
GO

EXEC dbo.sp_add_jobserver
    @job_name = N'Weekly Sales Backups',
    @server_name = N'SEATTLE2' ;
GO
```

For more information, see sp_add_jobserver (Transact-SQL).

See Also

Automated Administration Across an Enterprise

Feedback

Was this page helpful? Yes No

Monitor Job Activity

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

You can monitor the current activity of all defined jobs on an instance of SQL Server by using SQL Server Agent Job Activity Monitor.

SQL Server Agent Sessions

SQL Server Agent creates a new session each time the service starts. When a new session is created, the **sysjobactivity** table in the **msdb** database is populated with all the existing defined jobs. This table preserves the last activity for jobs when SQL Server Agent is restarted. Each session records SQL Server Agent normal job activity from the start of the job to its finish. Information about these sessions is stored in the **syssessions** table of the **msdb** database.

Job Activity Monitor

The Job Activity Monitor allows you to view the **sysjobactivity** table by using SQL Server Management Studio. You can view all jobs on the server, or you can define filters to limit the number of jobs displayed. You can also sort the job information by clicking on a column heading in the **Agent Job Activity** grid. For example, when you select the **Last Run** column heading, you can view the jobs in the order that they were last run. Clicking the column heading again toggles the jobs in ascending and descending order based on their last run date.

Using the Job Activity Monitor you can perform the following tasks:

- Start and stop jobs.
- View job properties.
- View the history for a specific job.

- Refresh the information in the **Agent Job Activity** grid manually or set an automatic refresh interval by clicking **View refresh settings**.

Use the Job Activity Monitor when you want to find out what jobs are scheduled to run, the last outcome of jobs that have run during the current session, and to find out which jobs are currently running or idle. If the SQL Server Agent service fails unexpectedly, you can determine which jobs were in the middle of being executed by looking at the previous session in the Job Activity Monitor.

To open the Job Activity Monitor, expand **SQL Server Agent** in Management Studio Object Explorer, right-click **Job Activity Monitor**, and click **View Job Activity**.

You can also view job activity for the current session by using the stored procedure **sp_help_jobactivity**.

Related Tasks

Expand table

Description	Topic
Describes how to view the runtime state of SQL Server Agent jobs.	View Job Activity

See Also

View Job Activity
sysjobactivity (Transact-SQL)
syssessions (Transact-SQL)
sp_help_jobactivity (Transact-SQL)

Feedback

Was this page helpful? Yes No

New Job Schedule - Job Schedule Properties

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to view and change the properties of the schedule.

Options

Name
Type a new name for the schedule.

Jobs in schedule
View the jobs that use the schedule.

Schedule type
Select the type of schedule.

Enabled
Check to enable or disable the schedule.

Recurring Schedule Types Options

Occurs
Select the interval at which the schedule recurs.

Recurs every
Select the number of days or weeks between recurrences of the schedule. This option is not available for monthly recurring schedules.

Monday
Set the job to occur on a Monday. Only available for weekly recurring schedules.

Tuesday
Set the job to occur on a Tuesday. Only available for weekly recurring schedules.

Wednesday
Set the job to occur on a Wednesday. Only available for weekly recurring schedules.

Thursday
Set the job to occur on a Thursday. Only available for weekly recurring schedules.

Friday
Set the job to occur on a Friday. Only available for weekly recurring schedules.

Saturday
Set the job to occur on a Saturday. Only available for weekly recurring schedules.

Sunday
Set the job to occur on a Sunday. Only available for weekly recurring schedules.

Day
Select the day of the month the schedule occurs. Only available for monthly recurring schedules.

of every
Select the number of months between occurrences of the schedule. Only available for monthly recurring schedules.

The
Specify a schedule for a specific day of the week on a specific week within the month. Only available for monthly recurring schedules.

Occurs once at
Set the time for a job to occur daily.

Occurs every
Set the number of hours, minutes, or seconds between occurrences.

Start date
Set the date when this schedule will become effective.

End date
Set the date when the schedule will no longer be effective.

No end date
Specify that the schedule will remain effective indefinitely.

One Time Schedule Types Options

Date
Select the date for the job to run.

Time
Select the time for the job to run.

See Also

Create and Attach Schedules to Jobs
Schedule a Job

Feedback

Was this page helpful? Yes No

Notify an Operator of Job Status

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to set notification options in SQL Server by using SQL Server Management Studio, Transact-SQL, or SQL Server Management Objects, so Microsoft SQL Server Agent can send notifications to operators about jobs.

Before You Begin

Security

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To notify an operator of job status

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**, expand **Jobs**, right-click the job you want to edit, and select **Properties**.
3. In the **Job Properties** dialog box, select the **Notifications** page.
4. If you want to notify an operator by e-mail, check **E-mail**, select an operator from the list, and then select one of the following:
    - **When the job succeeds** to notify the operator when the job completes successfully.

- **When the job fails** to notify the operator when the job completes unsuccessfully.
- **When the job completes** to notify the operator regardless of completion status.

5. If you want to notify an operator by pager, check **Page**, select an operator from the list, and then select one of the following:
   - **When the job succeeds** to notify the operator when the job completes successfully.
   - **When the job fails** to notify the operator when the job completes unsuccessfully.
   - **When the job completes** to notify the operator regardless of completion status.
6. If you want to notify an operator by net send, check **Net send**, select an operator from the list, and then select one of the following:
   - **When the job succeeds** to notify the operator when the job completes successfully.
   - **When the job fails** to notify the operator when the job completes unsuccessfully.
   - **When the job completes** to notify the operator regardless of completion status.

Using Transact-SQL

To notify an operator of job status

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- adds an e-mail notification for the specified alert (Test Alert).
-- This example assumes that Test Alert already exists
--  and that François Ajenstat is a valid operator name.
```

```
USE msdb ;
GO
EXEC dbo.sp_add_notification
@alert_name = N'Test Alert',
@operator_name = N'François Ajenstat',
@notification_method = 1 ;
GO
```

For more information, see sp_add_notification (Transact-SQL).

Using SQL Server Management Objects

To notify an operator of job status

Use the **Job** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell. For more information, see SQL Server Management Objects (SMO).

Feedback

Was this page helpful? Yes No

Operators

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Operators are aliases for people or groups that can receive electronic notification when jobs have completed or alerts have been raised. The SQL Server Agent service supports the notification of administrators through operators. Operators enable notification and monitoring capabilities of SQL Server Agent.

Operator Attributes and Concepts

The primary attributes of an operator are:

- Operator name
- Contact information

Naming an Operator

Every operator must have a name. Operator names must be unique within the SQL Server instance and can be no longer than **128** characters.

Contact Information

An operator's contact information defines how the operator is notified. Operators can be notified by e-mail, pager, or through the **net send** command:

> **Important**
>
> The Pager and **net send** options will be removed from SQL Server Agent in a future version of Microsoft SQL Server. Avoid using these features in new development work, and plan to modify applications that currently use these features.

- **E-mail notification**

  E-mail notification sends an e-mail message to the operator. For e-mail notification, you provide the e-mail address for the operator.

- **Pager notification**

  Paging is implemented by e-mail. For pager notification, you provide the e-mail address where the operator receives pager messages. To set up pager notification, you must install software on the mail server that processes inbound mail and converts it to a pager message. The software can take one of several approaches, including:

  - Forwarding the mail to a remote mail server at the site of the pager provider.

    The pager provider must offer this service, although the software required is generally available as part of the local mail system. For more information, see your pager documentation.

  - Routing the e-mail by way of the Internet to an e-mail server at the pager provider's site.

    This is a variation on the first approach.

  - Processing the inbound e-mail and dialing the pager by using an attached modem.

    This software is proprietary to pager service providers. The software acts as a e-mail client that periodically processes its inbox either by interpreting all or part of the e-mail address information as a pager number, or by matching the e-mail name to a pager number in a translation table.

    If all of the operators share a pager provider, you can use SQL Server Management Studio to specify any special e-mail formatting that is required by the pager-to-e-mail system. The special formatting can be a prefix or a suffix and can be included in the following lines of the e-mail:

    Subject:

    Cc:

    To:

    > ⓘ **Note**

If you use a low-capacity alphanumeric paging system, you can shorten the text that is sent by excluding the error text from the pager notification. An example of a low-capacity alphanumeric paging system is one that is limited to 64 characters per page.

- **net sendnotification**

  This sends a message to the operator by means of the **net send** command. For **net send**, specify the recipient (computer or user) of a network message.

  > ⓘ **Note**
  >
  > The **net send** command uses Microsoft Windows Messenger. To successfully send alerts, this service must be running on both the computer on which SQL Server is running and the computer that the operator uses.

Alerting and Fail-Safe Operators

You can choose which operators are to be notified in response to an alert. For example, you can assign rotating responsibilities for operator notification by scheduling alerts. For example, Individual A is notified of alerts that occur on Monday, Wednesday, or Friday, and Individual B is notified of alerts that occur on Tuesday, Thursday, or Saturday.

The fail-safe operator receives an alert notification after all pager notifications to the designated operators have failed. For example, if you have defined three operators for pager notifications and none of the designated operators can be paged, the fail-safe operator is notified.

The fail-safe operator is notified when:

- The operators responsible for the alert could not be paged.

  Reasons for failure to reach primary operators include incorrect pager addresses and off-duty operators.

- SQL Server Agent cannot access system tables in the **msdb** database.

  The **sysnotifications** system table specifies operator responsibilities for alerts.

The fail-safe operator is a security feature. You cannot delete the operator assigned to fail-safe duty without reassigning fail-safe duty to another operator, or deleting the fail-safe assignment altogether.

Notifying an Operator

You must set up one or more of the following in order to notify an operator:

- To send e-mail with Database Mail functionality, you must have access to an e-mail server that supports SMTP.
- For paging, you must have third-party pager-to-e-mail software and/or hardware.
- To use **net send**, the operator must be logged on to the specified computer, and the specified computer must allow messages from Windows Messenger.

Related Tasks

Expand table

Tasks	Topic
Tasks related to creating an operator	Create an Operator Designate a Fail-Safe Operator
Tasks related to assigning alerts	Assign Alerts to an Operator Define the Response to an Alert (SQL Server Management Studio) sp_add_notification (Transact-SQL) Assign Alerts to an Operator

See Also

Database Mail

Feedback

Was this page helpful? Yes No

Organize Jobs

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Job categories help you organize your jobs for easy filtering and grouping. For example, you can organize all your database backup jobs in the Database Maintenance category. You can also create your own job categories.

> **Warning**
>
> Multiserver categories exist only on a master server. There is only one default job category available on a master server: [**Uncategorized (Multi-Server)**]. When a multiserver job is downloaded, its category is changed to **Jobs From MSX** at the target server.

Related Tasks

Expand table

Description	Topic
Describes how to create a job category.	Create a Job Category
Describes how to delete a job category.	Delete a Job Category
Describes how to assign a job to a job category.	Assign a Job to a Job Category
Describes how to change the membership of a job category.	Change the Membership of a Job Category
Describes how to list the category information.	List Job Category Information

Feedback

Was this page helpful? Yes No

Pick Schedule for Job

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance differences from SQL Server for details.

Use this dialog to pick an existing schedule for the Microsoft SQL Server Agent job.

Options

Available schedules
Lists the schedules available for this job. Because a job and a schedule must have the same owner, this list only includes schedules owned by the owner of the job.

Name
Displays the name of the schedule.

Enabled
Selected if the schedule is enabled.

Description
Describes the conditions under which the schedule runs the job.

Jobs in schedule
Lists the job numbers attached to the schedule. Click a number to view the properties of the job.

Properties
Launches the **Job Schedule Properties** dialog where you can view information about the schedule.

See Also

Create and Attach Schedules to Jobs

Feedback

Was this page helpful? Yes No

Poll servers

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

When multiserver administration is implemented, target servers periodically contact the master server to upload information on jobs that have been executed, and download new jobs. The process of contacting the master server is called *server polling,* which takes place at regular *polling intervals*.

Polling intervals

The polling interval (one minute by default) controls how frequently the target server connects to the master server to download instructions and upload the results of job execution.

When a target server polls the master server, it reads the operations assigned to the target server from the `dbo.sysdownloadlist` table in the `msdb` database. These operations control multiserver jobs and various aspects of the behavior of a target server. Examples of operations include deleting a job, inserting a job, starting a job, and updating the polling interval of a target server.

Operations are posted to the `dbo.sysdownloadlist` table in either of the following ways:

- Explicitly by using the `dbo.sp_post_msx_operation` stored procedure.
- Implicitly by using other job stored procedures.

If you use job stored procedures to modify multiserver job schedules or job steps, or SQL Distributed Management Objects (SQL-DMO) to control multiserver jobs, issue the following command after modifying a multiserver job's steps or schedules:

SQL

```
EXECUTE msdb.dbo.sp_post_msx_operation 'INSERT', 'JOB', '<job id>'
```

Issue this command keeps the target servers synchronized with the current job definition.

If you use the following items, you don't have to post operations explicitly:

- Microsoft SQL Server Management Studio to control multiserver jobs.
- Job stored procedures that don't modify job schedules or job steps.

Force a target server to poll the master server

- SQL Server Management Studio
- Transact-SQL

See also

- Manage Events

Feedback

Was this page helpful? Yes No

Proxy Editor - Add Principal

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to grant server principals access to a Microsoft SQL Server Agent proxy account.

Options

Principal type
Type of principal to display.

Available Principals
Lists the principals of the type chosen.

Name
Select the name of the principal to grant access to that principal, or select **Name** to grant access to all principals in the list.

See Also

Create a SQL Server Agent Proxy

Feedback

Was this page helpful? Yes No

Remove Steps from a SQL Server Agent Master Job

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to remove steps from a SQL Server Agent master job in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Limitations and Restrictions

A SQL Server Agent master job cannot be targeted at both local and remote servers.

Security

Permissions

Unless you are a member of the **sysadmin** fixed server role, you can only modify jobs that you own. For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To remove steps from a SQL Server Agent master job

1. In **Object Explorer,** click the plus sign to expand the server that contains the job where you want to delete steps.
2. Click the plus sign to expand **SQL Server Agent**.

3. Click the plus sign to expand the **Jobs** folder.
4. Right-click the job where you want to delete steps and select **Properties**.
5. In the **Job Properties** *-job_name* dialog box, under **Select a page**, select **Steps**.
6. Under **Job step list**, select the job step you want to delete and click **Delete**.
7. When finished, click **OK**.

Using Transact-SQL

To remove steps from a SQL Server Agent master job

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- removes job step 1 from the job Weekly Sales Data Backup
USE msdb ;
GO

EXEC dbo.sp_delete_jobstep
    @job_name = N'Weekly Sales Data Backup',
    @step_id = 1 ;
GO
```

For more information, see sp_delete_jobstep (Transact-SQL).

Feedback

Was this page helpful? Yes No

Resize the job history log

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This article describes how to set size limits for Microsoft SQL Server Agent job history logs by using SQL Server Management Studio.

This feature doesn't work for SQL Server on Linux.

Security

For detailed information, see Implement SQL Server Agent Security.

Use SQL Server Management Studio

Resize the job history log based on raw size

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Right-click **SQL Server Agent**, and then select **Properties**.
3. Select the **History** page, and then confirm that **Limit size of job history log** is checked.
4. In the **Maximum job history log size** box, enter the maximum number of rows the job history log should allow.
5. In the **Maximum job history rows per job** box, enter the maximum number of job history rows to allow for a job.

Resize the job history log based on time

1. In **Object Explorer**, connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Right-click **SQL Server Agent**, and then select **Properties**.
3. Select the **History** page, and then select **Remove agent history**.
4. Select the appropriate number of **Days**, **Weeks**, or **Months**.

Feedback

Was this page helpful? Yes No

Run Jobs

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

To manage SQL Server Agent jobs, you can use SQL Server Management Studio, Transact-SQL stored procedures, or SQL Server Management Objects.

Related Tasks

Expand table

Description	Topic
Describes how to start running a Microsoft SQL Server Agent job.	Start a Job
Describes how to stop a Microsoft SQL Server Agent job.	Stop a Job
Describes how to disable or enable a Microsoft SQL Server Agent job.	Disable or Enable a Job

See Also

sysdownloadlist

Feedback

Was this page helpful? Yes No

Configure schedule for SQL Server Agent job

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

This article describes how to configure a schedule for a job for the SQL Server Agent to run on SQL Server or Azure SQL Managed Instance. Configure a SQL Server Agent job schedule by using SQL Server Management Studio (SSMS), Transact-SQL, or SQL Server Management Objects (SMO).

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Security

For detailed information, see Implement SQL Server Agent Security.

Prerequisites

To configure a schedule for a job, you should have already created a job. If you need to create a job, see Create a Job.

Configure schedule for a job

You can configure a schedule for a SQL Agent job by using SQL Server Management Studio (SSMS), Transact-SQL, or SQL Server Management Objects (SMO).

SQL Server Management Studio (SSMS)

To create a schedule in SQL Server Management Studio, follow these steps:

1. Open SQL Server Management Studio (SSMS).

2. In **Object Explorer,** connect to an instance of the SQL Server, and then expand that instance.
3. Expand **SQL Server Agent**, expand **Jobs**, right-click the job you want to schedule, and select **Properties**.
4. Select the **Schedules** page, and then select **New**.
5. In the **Name** box, type a name for the new schedule.
6. Clear the **Enabled** check box if you don't want the schedule to take effect immediately following its creation.
7. For **Schedule Type**, select one of the following:
    - Select **Start automatically when SQL Server Agent starts** to start the job when the SQL Server Agent service is started.
    - Select **Start whenever the CPUs become idle** to start the job when the CPUs reach an idle condition.
    - Select **Recurring** if you want a schedule to run repeatedly. To set the recurring schedule, complete the **Frequency**, **Daily Frequency**, and **Duration** groups on the dialog.
    - Select **One time** if you want the schedule to run only once. To set the **One time** schedule, complete the **One-time occurrence** group on the dialog.

To attach a schedule to a job in SSMS, follow these steps:

1. In **Object Explorer,** connect to an instance of SQL Server, and then expand that instance.
2. Expand **SQL Server Agent**, expand **Jobs**, right-click the job that you want to schedule, and select **Properties**.
3. Select the **Schedules** page, and then select **Pick**.
4. Select the schedule that you want to attach, and then select **OK**.
5. In the **Job Properties** dialog box, double-click the attached schedule.
6. Verify that **Start date** is set correctly. If it isn't, set the date when you want for the schedule to start, and then select **OK**.

7. In the **Job Properties** dialog box, select **OK**.

Related content

- Create a job
- SQL Server Agent

Feedback

Was this page helpful? Yes No

Select an account for the SQL Server Agent service

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

The service startup account defines the Microsoft Windows account in which SQL Server Agent runs and its network permissions. SQL Server Agent runs as a specified user account. You select an account for the SQL Server Agent service by using SQL Server Configuration Manager, where you can choose from the following options:

- **Built-in account**. You can choose from a list of the following built-in Windows service accounts:
  - **Local System** account. The name of this account is NT AUTHORITY\System. it's a powerful account that has unrestricted access to all local system resources. it's a member of the Windows **Administrators** group on the local computer.

    > **Important**
    >
    > The **Local System account** option is provided for backward compatibility only. The Local System account has permissions that SQL Server Agent doesn't require. Avoid running SQL Server Agent as the Local System account. For improved security, use a Windows domain account with the permissions listed in the following section, "Windows Domain Account Permissions."

- **This account**. Lets you specify the Windows domain account in which the SQL Server Agent service runs. We recommend choosing a Windows user account that isn't a member of the Windows **Administrators** group. However, there are limitations for using multiserver administration when the SQL Server Agent service account isn't a member of the local **Administrators** group. For more information, see 'Supported Service Account Types' that follows in this article.

Windows domain account permissions

For improved security, select **This account**, which specifies a Windows domain account. The Windows domain account that you specify must have the following permissions:

- In all Windows versions, permission to log on as a service (`SeServiceLogonRight`)

  > ⓘ **Note**
  >
  > The SQL Server Agent service account must be part of the Pre-Windows 2000 Compatible Access group on the domain controller, or jobs that are owned by domain users who aren't members of the Windows Administrators group fails.

- In Windows servers, the account that the SQL Server Agent Service runs as requires the following permissions can support SQL Server Agent proxies.
  - Permission to bypass traverse checking (`SeChangeNotifyPrivilege`)
  - Permission to replace a process-level token (`SeAssignPrimaryTokenPrivilege`)
  - Permission to adjust memory quotas for a process (`SeIncreaseQuotaPrivilege`)
  - Permission to access this computer from the network (`SeNetworkLogonRight`)

If the account doesn't have the permissions required to support proxies, only members of the **sysadmin** fixed server role can create jobs.

To receive Windows Management Instrumentation (WMI) alert notification, the service account for SQL Server Agent must have been granted permission to the namespace that contains the WMI events, and `ALTER ANY EVENT NOTIFICATION`.

SQL Server role membership

By default, the SQL Server Agent service account is mapped to the default SQL Server Agent service SID (`NT SERVICE\SQLSERVERAGENT`), which is a member of the **sysadmin** fixed server role. The account must also be a member of the `msdb` database role **TargetServersRole** on the master server if multiserver job processing is used. The Master Server Wizard automatically adds the service account to this role as part of the enlistment process.

Supported service account types

The following table lists the Windows account types that can be used for the SQL Server Agent service.

Expand table

Service account type	Nonclustered Server	Clustered server	Domain controller (nonclustered)
Microsoft Windows domain account (member of Windows Administrators group)	Supported	Supported	Supported
Windows domain account (non-administrative)	Supported See Limitation 1 later in this section.	Supported See Limitation 1 later in this section.	Supported See Limitation 1 later in this section.
Network Service account (`NT AUTHORITY\NetworkService`)	Supported See Limitation 1, 3, and 4 later in this section.	Not supported	Not supported
Local user account (non-administrative)	Supported See Limitation 1 later in this section.	Not supported	Not applicable
Local System account (`NT AUTHORITY\System`)	Supported See Limitation 2 later in this section.	Not supported	Supported See Limitation 2 later in this section.
Local Service account (`NT AUTHORITY\LocalService`)	Not supported	Not supported	Not supported

Limitation 1: Using non-administrative accounts for multiserver administration

Enlisting target servers to a master server might fail with the following error message: `The enlist operation failed.`

To resolve this error, restart both the SQL Server and the SQL Server Agent services. For more information, see Start, stop, pause, resume, and restart SQL Server services.

Limitation 2: Using the Local System account for multiserver administration

Multiserver administration is supported when the SQL Server Agent service is run under the Local System account only when both the master server and the target server reside on the same computer. If you use this configuration, the following message is returned when you enlist target servers to the master server:

Output

```
Ensure the agent start-up account for <target_server_computer_name> has
rights to log on as targetServer.
```

You can ignore this informational message. The enlistment operation should complete successfully. For more information, see Create a multiserver environment.

Limitation 3: Using the Network Service account when it's a SQL Server user

The SQL Server Agent might fail to start if you run the SQL Server Agent service under the Network Service account, and the Network Service account is explicitly granted access to log into a SQL Server instance as a SQL Server user.

To resolve this, restart the computer where SQL Server is running. This only needs to be done once.

Limitation 4: Using the Network Service account when SQL Server Reporting Services is running on the same computer

The SQL Server Agent might fail to start if you run the SQL Server Agent service under the Network Service account and Reporting Services is also running on the same computer.

To resolve this, restart the computer where SQL Server is running, and then restart both the SQL Server and the SQL Server Agent services. This only needs to be done once.

Common tasks

- Set the Service Startup Account for SQL Server Agent (SQL Server Configuration Manager)

- Configure SQL Server Agent mail to use Database Mail

Use SQL Server Configuration Manager to specify that SQL Server Agent must start up when the operating system starts.

Related content

- Configure Windows service accounts and permissions
- SQL Server Configuration Manager: Connect to another computer
- Implement SQL Server Agent Security

Feedback

Was this page helpful? Yes No

Send SQL Server Agent Error Messages

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to how to configure Microsoft SQL Server Agent to send its error messages by way of net send in SQL Server by using SQL Server Management Studio.

Before You Begin

Limitations and Restrictions

- Object Explorer only displays the SQL Server Agent node if you have permission to use it.
- The Microsoft Windows Messenger service must be running to receive net send events.

Security

Permissions

To perform its functions, SQL Server Agent must be configured to use the credentials of an account that is a member of the **sysadmin** fixed server role in SQL Server. The account must have the following Windows permissions:

- Log on as a service (SeServiceLogonRight)
- Replace a process-level token (SeAssignPrimaryTokenPrivilege)
- Bypass traverse checking (SeChangeNotifyPrivilege)
- Adjust memory quotas for a process (SeIncreaseQuotaPrivilege)

For more information about the Windows permissions required for the SQL Server Agent service account, see Select an Account for the SQL Server Agent Service and Setting Up Windows Service Accounts.

Using SQL Server Management Studio

To send SQL Server Agent error messages

1. In **Object Explorer**, click the plus sign to expand the server that contains the SQL Server Agent error log from which you want to send error messages via net send.
2. Right-click **SQL Server Agent** and select **Properties**.
3. In the **SQL Server Agent Properties -***server_name* dialog box, under **Error log** on the **General** page, , type the user name or computer name to which you want to send error messages in the **Net send recipient** box.
4. Click **OK**.

Feedback

Was this page helpful? Yes No

Specify a Target Server's Location

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to specify the location of a target server in a multiserver administration configuration in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Limitations and Restrictions

Performing this action edits the registry. Manual editing of the registry is not recommended because inappropriate or incorrect changes can cause serious configuration problems for your system. Therefore, only experienced users should use the Registry Editor program to edit the registry. For more information, see the Microsoft Windows documentation.

Security

Permissions

Requires membership in the **sysadmin** fixed server role.

Using SQL Server Management Studio

To specify a target server's location

1. In **Object Explorer**, click the plus sign to expand the master server on which you want to specify a target server's location.

2. Right-click **SQL Server Agent**, point to **Multi Server Administration**, and select **Manage Target Servers**.

3. Right-click a target server and select **Properties**.

4. In the **Location** box, enter a location for the server, and then click **OK**.

Using Transact-SQL

To specify a target server's location

1. In **Object Explorer**, connect to an instance of Database Engine.

2. On the Standard bar, click **New Query**.

3. Copy and paste the following example into the query window and click **Execute**.

```
USE msdb ;
GO
-- enlists the current server into the AdventureWorks1 master server.
-- The location for the current server is Building 21, Room 309, Rack 5
EXEC dbo.sp_msx_enlist N'AdventureWorks2022',
    N'Building 21, Room 309, Rack 5' ;
GO
```

For more information, see sp_msx_enlist (Transact-SQL).

Feedback

Was this page helpful? Yes No

Specify Job Responses

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Job responses specify actions that the SQL Server Agent service will take after a job completes. Job responses ensure that database administrators know when jobs complete and how frequently they run. Typical job responses include:

- Notifying the operator by using e-mail, electronic paging, or a **net send** message.

  Use one of these job responses if the operator must perform a follow-up action. For example, if a backup job completes successfully, the operator must be notified to remove the backup tape and store it in a safe location.

- Writing an event message to the Windows application log.

  You can use this response only for failed jobs.

- Automatically deleting the job.

  Use this job response if you are certain that you do not need to rerun this job.

Related Tasks

Expand table

Description	Topic
Describes how to notify an operator of job status.	Notify an Operator of Job Status
Describes how to write job status to the Windows application log.	Write the Job Status to the Windows Application Log

See Also

Feedback

Was this page helpful? Yes No

SQL Server Agent fixed database roles

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

SQL Server has the following `msdb` database fixed database roles, which give administrators finer control over access to SQL Server Agent. The roles listed from least to most privileged access are:

- **SQLAgentUserRole**
- **SQLAgentReaderRole**
- **SQLAgentOperatorRole**

When users who aren't members of one of these roles are connected to SQL Server in SQL Server Management Studio, the **SQL Server Agent** node in Object Explorer isn't visible. A user must be a member of one of these fixed database roles or a member of the **sysadmin** fixed server role to use SQL Server Agent.

Permissions of SQL Server Agent fixed database roles

The SQL Server Agent database role permissions are concentric in relation to one another. In other words, more privileged roles inherit the permissions of less privileged roles on SQL Server Agent objects (including alerts, operators, jobs, schedules, and proxies). For example, if members of least-privileged **SQLAgentUserRole** have been granted access to `proxy_A`, members of both **SQLAgentReaderRole** and **SQLAgentOperatorRole** automatically have access to this proxy even though access to `proxy_A` hasn't been explicitly granted to them. This might have security implications, which are discussed in the following sections about each role.

SQLAgentUserRole permissions

SQLAgentUserRole is the least privileged of the SQL Server Agent fixed database roles. It has permissions on only operators, local jobs, and job schedules. Members of **SQLAgentUserRole** have permissions on only local jobs and job schedules that they own. They can't use multiserver jobs (master and target server jobs), and they can't change job ownership to gain access to jobs that they don't already own. **SQLAgentUserRole** members can view a list of available proxies only in the **Job Step Properties** dialog box of SQL Server Management Studio. Only the **Jobs** node in SQL Server Management Studio Object Explorer is visible to members of **SQLAgentUserRole**.

> **Important**
>
> The **SQLAgentReaderRole** and the **SQLAgentOperatorRole** are automatically members of the **SQLAgentUserRole**. This means that members of **SQLAgentReaderRole** and **SQLAgentOperatorRole** have access to all SQL Server Agent proxies that have been granted to the **SQLAgentUserRole** and can use those proxies.

The following table summarizes **SQLAgentUserRole** permissions on SQL Server Agent objects.

Expand table

Action	Operators	Local jobs (owned jobs only)	Job schedules (owned schedules only)	Proxies
Create/modify/delete	No	Yes Can't change job ownership.	Yes	No
View list (enumerate)	Yes Can get list of available operators for use in `sp_notify_operator` and the **Job Properties** dialog box of Management Studio.	Yes	Yes	Yes List of proxies only available in the **Job Step Properties** dialog box of Management Studio.
Enable/disable	No	Yes	Yes	Not applicable
View properties	No	Yes	Yes	No
Execute/stop/start	Not applicable	Yes	Not applicable	Not applicable
View job history	Not applicable	Yes	Not applicable	Not applicable
Delete job history	Not applicable	No Members of **SQLAgentUserRole** must explicitly be granted the `EXECUTE` permission on `sp_purge_jobhistory` to delete job history on jobs that they own. They can't delete job history for any other jobs.	Not applicable	Not applicable
Attach/detach	Not applicable	Not applicable	Yes	Not applicable

SQLAgentReaderRole permissions

SQLAgentReaderRole includes all the **SQLAgentUserRole** permissions, as well as permissions to view the list of available multiserver jobs, their properties, and their history. Members of this role can also view the list of all available jobs and job schedules and their properties, not just those jobs and job schedules that they own. **SQLAgentReaderRole** members can't change job ownership to gain access to jobs that they don't already own. Only the **Jobs** node in SQL Server Management Studio Object Explorer is visible to members of the **SQLAgentReaderRole**.

> **Important**
>
> Members of **SQLAgentReaderRole** are automatically members of the **SQLAgentUserRole**. This means that members of **SQLAgentReaderRole** have access to all SQL Server Agent proxies that have been granted to **SQLAgentUserRole** and can use those proxies.

The following table summarizes **SQLAgentReaderRole** permissions on SQL Server Agent objects.

Expand table

Action	Operators	Local jobs	Multiserver jobs	Job schedules	Proxies
Create/modify/delete	No	Yes (owned jobs only) Can't change job ownership.	No	Yes (owned schedules only)	No
View list (enumerate)	Yes Can get list of available operators for use in `sp_notify_operator` and the **Job Properties** dialog box of Management Studio.	Yes	Yes	Yes	Yes List of proxies only available in the **Job Step Properties** dialog box of Management Studio.
Enable/disable	No	Yes (owned jobs only)	No	Yes (owned schedules only)	Not applicable
View properties	No	Yes	Yes	Yes	No
Edit properties	No	Yes (owned jobs only)	No	Yes (owned schedules only)	No
Execute/stop/start	Not applicable	Yes (owned jobs only)	No	Not applicable	Not applicable
View job history	Not applicable	Yes	Yes	Not applicable	Not applicable
Delete job history	Not applicable	No Members of **SQLAgentReaderRole** must explicitly be granted the `EXECUTE` permission on `sp_purge_jobhistory` to delete job history on jobs that they own. They can't delete job history for any other jobs.	No	Not applicable	Not applicable
Attach/detach	Not applicable	Not applicable	Not applicable	Yes (owned schedules only)	Not applicable

SQLAgentOperatorRole permissions

SQLAgentOperatorRole is the most privileged of the SQL Server Agent fixed database roles. It includes all the permissions of **SQLAgentUserRole** and **SQLAgentReaderRole**. Members of this role can also view properties for operators and proxies, and enumerate available proxies and alerts on the server.

SQLAgentOperatorRole members have extra permissions on local jobs and schedules. They can execute, stop, or start all local jobs, and they can delete the job history for any local job on the server. They can also enable or disable all local jobs and schedules on the server. To enable or disable local jobs or schedules, members of this role must use the stored procedures `sp_update_job` and `sp_update_schedule`. Only the parameters that specify the job or

schedule name or identifier and the `@enabled` parameter can be specified by members of **SQLAgentOperatorRole**. If they specify any other parameters, execution of these stored procedures fails. **SQLAgentOperatorRole** members can't change job ownership to gain access to jobs that they don't already own.

The **Jobs**, **Alerts**, **Operators**, and **Proxies** nodes in SQL Server Management Studio Object Explorer are visible to members of **SQLAgentOperatorRole**. Only the **Error Logs** node isn't visible to members of this role.

> **Important**
>
> Members of **SQLAgentOperatorRole** are automatically members of **SQLAgentUserRole** and **SQLAgentReaderRole**. This means that members of **SQLAgentOperatorRole** have access to all SQL Server Agent proxies that have been granted to either **SQLAgentUserRole** or **SQLAgentReaderRole** and can use those proxies.

The following table summarizes **SQLAgentOperatorRole** permissions on SQL Server Agent objects.

Expand table

Action	Alerts	Operators	Local jobs	Multiserver jobs	Job schedules	Proxies
Create/modify/delete	No	No	Yes (owned jobs only) Can't change job ownership.	No	Yes (owned schedules only)	No
View list (enumerate)	Yes	Yes Can get list of available operators for use in `sp_notify_operator` and the **Job Properties** dialog box of Management Studio.	Yes	Yes	Yes	Yes
Enable/disable	No	No	Yes **SQLAgentOperatorRole** members can enable or disable local jobs they don't own by using the stored procedure `sp_update_job` and specifying values for the `@enabled` and the `@job_id` (or `@job_name`) parameters. If a member of this role specifies any other parameters for this stored procedure, execution of the procedure fails.	No	Yes **SQLAgentOperatorRole** members can enable or disable schedules they don't own by using the stored procedure `sp_update_schedule` and specifying values for the `@enabled` and the `@schedule_id` (or `@name`) parameters. If a member of this role specifies any other parameters for this stored procedure, execution of the procedure fails.	Not applicable
View properties	Yes	Yes	Yes	Yes	Yes	Yes

Action	Alerts	Operators	Local jobs	Multiserver jobs	Job schedules	Proxies
Edit properties	No	No	Yes (owned jobs only)	No	Yes (owned schedules only)	No
Execute/stop/start	Not applicable	Not applicable	Yes	No	Not applicable	Not applicable
View job history	Not applicable	Not applicable	Yes	Yes	Not applicable	Not applicable
Delete job history	Not applicable	Not applicable	Yes	No	Not applicable	Not applicable
Attach/detach	Not applicable	Not applicable	Not applicable	Not applicable	Yes (owned schedules only)	Not applicable

Assign multiple roles to users

Members of the **sysadmin** fixed server role have access to all SQL Server Agent functionality. If a user isn't a member of the **sysadmin** role, but is a member of more than one SQL Server Agent fixed database role, it's important to remember the concentric permissions model of these roles. Because more privileged roles always contain all the permissions of less privileged roles, a user who is a member of more than one role automatically has the permissions associated with the most privileged role that the user is a member of.

Related content

- Implement SQL Server Agent Security
- sp_update_job (Transact-SQL)
- sp_update_schedule (Transact-SQL)
- sp_notify_operator (Transact-SQL)
- sp_purge_jobhistory (Transact-SQL)

Feedback

Was this page helpful? Yes No

Synchronize Target Server Clocks

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to synchronize target server clocks in SQL Server with the master server clock by using SQL Server Management Studio or Transact-SQL. Synchronizing these system clocks supports your job schedules.

Before You Begin

Security

Permissions

Requires membership in the **sysadmin** fixed server role.

Using SQL Server Management Studio

To synchronize target server clocks

1. In **Object Explorer,** click the plus sign to expand the server where you want to synchronize the target server clocks with the master server clock.
2. Right-click **SQL Server Agent**, point to **Multi Server Administration**, and select **Manage Target Servers**.
3. In the **Manage Target Servers** dialog box, click **Post Instructions**.
4. In the **Instruction type** list, select **Synchronize clocks**.
5. Under **Recipients**, do one of the following:

- Click **All target servers** to synchronize all target server clocks with the master server clock.
- Click **These target servers** to synchronize certain server clocks, and then select each target server whose clock you want to synchronize with the master server clock.

6. When finished, click **OK**.

Using Transact-SQL

To synchronize target server clocks

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
USE msdb ;
GO
-- resynchronizes the SEATTLE1 target server
EXEC dbo.sp_resync_targetserver
    N'SEATTLE1' ;
GO
```

For more information, see sp_resync_targetserver (Transact-SQL).

Feedback

Was this page helpful? Yes No

Troubleshoot Multiserver Jobs That Use Proxies

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Distributed jobs whose steps are associated with a proxy run under the context of the proxy account on the target server. If job steps that use proxy accounts fail when downloaded from the master server, check the **error_message** column in the **sysdownloadlist** table in the **msdb** database for the following error messages:

- "The job step requires a proxy account, however proxy matching is disabled on the target server."

  To resolve this error, set the **\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.***<n>***\SQLServerAgent\AllowDownloadedJobsToMatchProxyName** registry subkey to **1 (true)**. By default, this subkey is set to **0** (**false**). The value of **MSSQL.***<n>* is the instance name; for example, **MSSQL.1** or **MSSQL.3**.

- "Proxy not found."

  To resolve this error, make sure a proxy account exists on the target server with the same name as the master server proxy account under which the job step runs.

> **Caution**
>
> Incorrectly editing the registry can severely damage your system. Before making changes to the registry, we recommend that you back up any valued data on the computer.

See Also

Create a Multiserver Environment

Feedback

Was this page helpful? Yes No

Tune Automated Administration Across an Enterprise

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Multiserver administration with Microsoft SQL Server Agent takes advantage of the self-tuning features of SQL Server. Therefore, under normal conditions, additional job tuning is not necessary. However, network loads increase when you run jobs, generate alerts, and notify operators. You can tune automated administration across an enterprise to minimize the network traffic these activities cause.

Related content

- Monitoring Performance of the Data Flow Engine

Feedback

Was this page helpful? Yes No

Use Performance Objects

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

SQL Server Agent includes performance objects and counters to monitor how the service is performing. These performance objects allow you to use Performance Monitor, a Windows tool, to identify what the SQL Server Agent service is doing in the background. For example, you can identify how many active jobs the SQL Server Agent service is currently running to identify those jobs that are blocked.

The SQL Server Agent service performance objects and counters exist for each instance of SQL Server that is installed on a computer. Performance objects are named according to the instance of SQL Server that each object represents.

The following table shows how the SQL Server Agent service performance objects are named:

Expand table

Instance type	Object name
Default	**SQLAgent:***object*:*counter*
Named	**SQLAgent$** ***instance_name**** :*object*:*counter*

SQL Server includes the following performance objects for SQL Server Agent.

Expand table

Object name	Description
SQLAgent:Jobs	Performance information about jobs that have been started, success rates, and current status
SQLAgent:JobSteps	Status information about job steps

Object name	Description
SQLAgent:Alerts	Information about number of alerts and notifications
SQLAgent:Statistics	General performance information

See Also

Monitor and Tune for Performance
How to: Start System Monitor (Windows)

Feedback

Was this page helpful? Yes No

Use Tokens in Job Steps

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

SQL Server Agent allows you to use tokens in Transact-SQL job step scripts. Using tokens when you write your job steps gives you the same flexibility that variables provide when you write software programs. After you insert a token in a job step script, SQL Server Agent replaces the token at run time, before the job step is executed by the Transact-SQL subsystem.

Understand Using Tokens

> **Important**
>
> Any Windows user with write permissions on the Windows Event Log can access job steps that are activated by SQL Server Agent alerts or WMI alerts. To avoid this security risk, SQL Server Agent tokens that can be used in jobs activated by alerts are disabled by default. These tokens are: **A-DBN**, **A-SVR**, **A-ERR**, **A-SEV**, **A-MSG**, and **WMI(*property*)**. In this release, use of tokens is extended to all alerting.
>
> If you need to use these tokens, first ensure that only members of trusted Windows security groups, such as the Administrators group, have write permissions on the Event Log of the computer where SQL Server resides. Then, right-click **SQL Server Agent** in Object Explorer, select **Properties**, and on the **Alert System** page, select **Replace tokens for all job responses to alerts** to enable these tokens.

SQL Server Agent token replacement is simple and efficient: SQL Server Agent replaces an exact literal string value for the token. All tokens are case-sensitive. Your job steps must take this into account and correctly quote the tokens you use or convert the replacement string to the correct data type.

For example, you might use the following statement to print the name of the database in a job step:

```
PRINT N'Current database name is $(ESCAPE_SQUOTE(A-DBN))' ;
```

In this example, the **ESCAPE_SQUOTE** macro is inserted with the **A-DBN** token. At run time, the **A-DBN** token is replaced with the appropriate database name. The escape macro escapes any single quotation marks that may be inadvertently passed in the token replacement string. SQL Server Agent replaces one single quotation mark with two single quotation marks in the final string.

For example, if the string passed to replace the token is `AdventureWorks2022'SELECT @@VERSION --`, the command executed by the SQL Server Agent job step will be:

```
PRINT N'Current database name is AdventureWorks2022''SELECT @@VERSION --' ;
```

In this case, the inserted statement, `SELECT @@VERSION`, doesn't execute. Instead, the extra single quotation mark causes the server to parse the inserted statement as a string. If the token replacement string doesn't contain a single quotation mark, no characters are escaped and the job step containing the token executes as intended.

To debug token usage in your job steps, use print statements such as `PRINT N'$(ESCAPE_SQUOTE(SQLDIR))'` and save job step output to a file or table. Use the **Advanced** page of the **Job Step Properties** dialog box to specify a job step output file or table.

SQL Server Agent Tokens and Macros

The following tables list and describe the tokens and macros that SQL Server Agent supports.

SQL Server Agent Tokens

Expand table

Token	Description
(A-DBN)	Database name. If the job is run by an alert, the database name value automatically replaces this token in the job step.
(A-SVR)	Server name. If the job is run by an alert, the server name value automatically replaces this token in the job step.

Token	Description
(A-ERR)	Error number. If the job is run by an alert, the error number value automatically replaces this token in the job step.
(A-SEV)	Error severity. If the job is run by an alert, the error severity value automatically replaces this token in the job step.
(A-MSG)	Message text. If the job is run by an alert, the message text value automatically replaces this token in the job step.
(JOBNAME)	The name of the job. This token is only available on SQL Server 2016 and above.
(STEPNAME)	The name of the step. This token is only available on SQL Server 2016 and above.
(DATE)	Current date (in YYYYMMDD format).
(INST)	Instance name. For a default instance, this token has the default instance name: MSSQLSERVER.
(JOBID)	Job ID.
(MACH)	Computer name.
(MSSA)	Master SQLServerAgent service name.
(OSCMD)	Prefix for the program used to run **CmdExec** job steps.
(SQLDIR)	The directory in which SQL Server is installed. By default, this value is C:\Program Files\Microsoft SQL Server\MSSQL.
(SQLLOGDIR)	Replacement token for the SQL Server error log folder path - for example, $(ESCAPE_SQUOTE(SQLLOGDIR)). This token is only available on SQL Server 2014 and above.
(STEPCT)	A count of the number of times this step has executed (excluding retries). Can be used by the step command to force termination of a multistep loop.
(STEPID)	Step ID.
(SRVR)	Name of the computer running SQL Server. If the SQL Server instance is a named instance, this includes the instance name.
(TIME)	Current time (in HHMMSS format).
(STRTTM)	The time (in HHMMSS format) that the job began executing.
(STRTDT)	The date (in YYYYMMDD format) that the job began executing.
(WMI(*property*))	For jobs that run in response to WMI alerts, the value of the property specified by *property*. For example, `$(WMI(DatabaseName))` provides the value of the

Token	Description
	DatabaseName property for the WMI event that caused the alert to run.

SQL Server Agent Escape Macros

Expand table

Escape Macros	Description
$(ESCAPE_SQUOTE(*token_name***))**	Escapes single quotation marks (') in the token replacement string. Replaces one single quotation mark with two single quotation marks.
$(ESCAPE_DQUOTE(*token_name***))**	Escapes double quotation marks (") in the token replacement string. Replaces one double quotation mark with two double quotation marks.
$(ESCAPE_RBRACKET(*token_name***))**	Escapes right brackets (]) in the token replacement string. Replaces one right bracket with two right brackets.
$(ESCAPE_NONE(*token_name***))**	Replaces token without escaping any characters in the string. This macro is provided to support backward compatibility in environments where token replacement strings are only expected from trusted users. For more information, see "Updating Job Steps to Use Macros," later in this topic.

Update Job Steps to Use Macros

The following table describes how token replacement is handled by SQL Server Agent. To turn alert token replacement on or off, right-click **SQL Server Agent** in Object Explorer, select **Properties**, and on the **Alert System** page, select or clear the **Replace tokens for all job responses to alerts** check box.

Expand table

Token syntax	Alert token replacement on	Alert token replacement off
ESCAPE macro used	All tokens in jobs are successfully replaced.	Tokens activated by alerts aren't replaced. These tokens are **A-DBN**, **A-SVR**, **A-ERR**, **A-SEV**, **A-MSG**, and **WMI(***property***)**. Other static tokens are replaced successfully.

Token syntax	Alert token replacement on	Alert token replacement off
No ESCAPE macro used	Any jobs containing tokens fail.	Any jobs containing tokens fail.

Token Syntax Update Examples

Below are token syntax examples to help illustrate how to use these commands.

A. Use tokens in non-nested strings

The following example shows how to update a simple non-nested script with the appropriate escape macro. Before running the update script, the following job step script uses a job step token to print the appropriate database name:

```
PRINT N'Current database name is $(A-DBN)' ;
```

After running the update script, an `ESCAPE_NONE` macro is inserted before the `A-DBN` token. Because single quotation marks are used to delimit the print string, you must update the job step by inserting the `ESCAPE_SQUOTE` macro as follows:

```
PRINT N'Current database name is $(ESCAPE_SQUOTE(A-DBN))' ;
```

B. Use tokens in nested strings

In job step scripts where tokens are used in nested strings or statements, the nested statements should be rewritten as multiple statements before inserting the appropriate escape macros.

For example, consider the following job step, which uses the `A-MSG` token and hasn't been updated with an escape macro:

```
PRINT N'Print ''$(A-MSG)''' ;
```

After running the update script, an `ESCAPE_NONE` macro is inserted with the token. However, in this case, you would have to rewrite the script without using nesting as follows and insert the `ESCAPE_SQUOTE` macro to properly escape delimiters that may be passed in the token replacement string:

```
SQL
```

```
DECLARE @msgString nvarchar(max);
SET @msgString = '$(ESCAPE_SQUOTE(A-MSG))';
SET @msgString = QUOTENAME(@msgString,'''');
PRINT N'Print ' + @msgString;
```

> **Note**
>
> In this example the QUOTENAME function sets the quote character.

C. Use tokens with the ESCAPE_NONE macro

The following example is part of a script that retrieves the job_id from the sysjobs table and uses the JOBID token to populate the @JobID variable, which was declared earlier in the script as a binary data type.

> **Note**
>
> Because no delimiters are required for binary data types, the ESCAPE_NONE macro is used with the JOBID token. You wouldn't need to update this job step after running the update script.

SQL

```
DECLARE @JobID uniqueidentifier
SET @JobID = $(ESCAPE_NONE(JOBID))
```

This directly assigns the value from the JOBID token to @JobID, eliminating unnecessary database queries and implicit conversions, addressing the concerns raised in section C.

See also

- Implement Jobs
- Manage Job Steps

Feedback

Was this page helpful? Yes No

Write the Job Status to the Windows Application Log

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to configure Microsoft SQL Server Agent in SQL Server to write job status to the Windows application event log by using SQL Server Management Studio, Transact-SQL, or SQL Server Management Objects.

Job responses ensure that database administrators know when jobs complete and how frequently they run. Typical job responses include:

- Notifying the operator by using e-mail, electronic paging, or a **net send** message. Use one of these job responses if the operator must perform a follow-up action. For example, if a backup job completes successfully, the operator must be notified to remove the backup tape and store it in a safe location.
- Writing an event message to the Windows application log. You can use this response only for failed jobs.
- Automatically deleting the job. Use this job response if you are certain that you do not need to rerun this job.

Before You Begin

Security

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To write job status to the Windows application log

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**, expand **Jobs**, right-click the job you want to edit, and then click **Properties**.
3. Select the **Notifications** page.
4. Check **Write to Windows application event log**, and choose one of the following:
    - Click **When the job succeeds** to log the job status when the job completes successfully.
    - Click **When the job fails** to log the job status when the job completes unsuccessfully.
    - Click **When the job completes** to log the job status regardless of completion status.

Using SQL Server Management Objects

To write job status to the Windows application log

Call the **EventLogLevel** property of the **Job** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell.

The following code example sets the job to generate an operating system event log entry when the job execution finishes.

PowerShell

```
$srv = new-object Microsoft.SqlServer.Management.Smo.Server("(local)")
$jb = new-object
Microsoft.SqlServer.Management.Smo.Agent.Job($srv.JobServer, "Test Job")
$jb.EventLogLevel =
[Microsoft.SqlServer.Management.Smo.Agent.CompletionAction]::Always
```

Feedback

Was this page helpful? Yes No

Create a CmdExec Job Step

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to create and define a Microsoft SQL Server Agent job step in SQL Server that uses an executable program or operating system command by using SQL Server Management Studio, Transact-SQL or SQL Server Management Objects.

Before You Begin

Security

By default, only members of the **sysadmin** fixed server role can create CmdExec job steps. These job steps run under the context of the SQL Server Agent service account unless the **sysadmin** user creates a proxy account. Users who are not members of the **sysadmin** role can create CmdExec job steps if they have access to a CmdExec proxy account.

Permissions

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To create a CmdExec job step

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**, create a new job or right-click an existing job, and then click **Properties**.

3. In the **Job Properties** dialog, click the **Steps** page, and then click **New**.

4. In the **New Job Step** dialog, type a job **Step name**.

5. In the **Type** list, choose **Operating system (CmdExec)**.

6. In **Run as** list, select the proxy account with the credentials that the job will use. By default, CmdExec job steps run under the context of the SQL Server Agent service account.

7. In the **Process exit code of a successful command** box, enter a value from 0 to 999999.

8. In the **Command** box, enter the operating system command or executable program. See "Using Transact T-SQL for an example.

9. Click the **Advanced** page to set job step options, such as: what action to take if the job step succeeds or fails, how many times SQL Server Agent should try to execute the job step, and the file where SQL Server Agent can write the job step output. Only members of the **sysadmin** fixed server role can write job step output to an operating system file.

Using Transact-SQL

To create a CmdExec job step

1. In **Object Explorer**, connect to an instance of Database Engine.

2. On the Standard bar, click **New Query**.

3. Copy and paste the following example into the query window and click **Execute**.

```
-- creates a job step that uses CmdExec
USE msdb;
GO
EXEC sp_add_jobstep
    @job_name = N'Weekly Sales Data Backup',
    @step_name = N'Set database to read only',
    @subsystem = N'CMDEXEC',
    @command = 'C:\clickme_scripts\SQL11\PostBOLReorg GetHsX.exe',
    @retry_attempts = 5,
    @retry_interval = 5 ;
GO
```

For more information, see sp_add_jobstep (Transact-SQL)

Using SQL Server Management Objects

To create a CmdExec job step

Use the **JobStep** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell.

Feedback

Was this page helpful? Yes No

Create a Job Category

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This article describes how to create a job category in SQL Server by using SQL Server Management Studio, Transact-SQL or SQL Server Management Objects.

SQL Server Agent provides built-in job categories that you can assign jobs to, or you can create a job category and assign jobs to it. Job categories help you organize your jobs for easy filtering and grouping. For example, you can organize all your database backup jobs in the Database Maintenance category. You can also create your own job categories.

Before You Begin

Limitations and Restrictions

Multiserver categories exist only on a master server. There is only one default job category available on a master server: [**Uncategorized (Multi-Server)**]. When a multiserver job is downloaded, its category is changed to **Jobs From MSX** at the target server.

Security

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To create a job category

1. In **Object Explorer**, select the plus sign to expand the server where you want to create a job category.
2. Select the plus sign to expand **SQL Server Agent**.
3. Right-click the **Jobs** folder and select **Manage Job Categories**.
4. In the **Manage Job Categories***server_name* dialog box, select **Add**.
5. In the new dialog box, in the **Name** box, enter a name for the new job category.
6. Select the **Show all jobs** check box. Select one or more jobs for the new category by checking the boxes corresponding to the jobs.
7. Select **OK**.
8. In the **Manage Job Categories***server_name* dialog box, select **Refresh** to ensure that the new job category is active. If everything looks as expected, close this dialog box.

For more information on these dialog boxes, see Job Categories - Manage Job Categories and Job Categories Properties - New Job Category.

Using Transact-SQL

To create a job category

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, select **New Query**.
3. Copy and paste the following example into the query window and select **Execute**.

```SQL
-- creates a local job category named AdminJobs
USE msdb ;
GO
EXEC dbo.sp_add_category
    @class=N'JOB',
    @type=N'LOCAL',
    @name=N'AdminJobs' ;
GO
```

For more information, see sp_add_category (Transact-SQL).

Using SQL Server Management Objects

To create a job category

Call the **JobCategory** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell. For example code, see Scheduling Automatic Administrative Tasks in SQL Server Agent.

Related content

- Create a Job Step
- Create a Job

Feedback

Was this page helpful? Yes No

Create a SQL Server Agent Job in SQL Server Management Studio (SSMS)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most SQL Server Agent features are supported. See Azure SQL Managed Instance T-SQL differences for more details.

This article explains how to create a SQL Server Agent job using **SQL Server Management Studio (SSMS)**, **Transact-SQL (T-SQL)**, or **SQL Server Management Objects (SMO)**.

To add job steps, schedules, alerts, and notifications that can be sent to operators, see the links to topics in the See Also section.

Prerequisites

- User must be a member of SQL Server Agent fixed database roles or the **sysadmin** role.
- Only job owners or members of **sysadmin** can modify jobs.
- Assigning a job to another login does not guarantee sufficient permissions to run the job.

Security Considerations

- Only **sysadmin** can change the job owner.
- **Sysadmin** can assign job ownership to other users and run any job.
- Jobs with steps requiring proxy accounts need to ensure the new owner has access to those proxies, or the job will fail.

For detailed security information, see Implement SQL Server Agent Security

How to Create a Job using SSMS

1. In **Object Explorer**, expand the server where the job will be created.
2. Expand **SQL Server Agent**.

3. Right-click **Jobs** and select **New Job...**.
4. On the **General** page, configure job properties. For more details, see Job Properties - General Page.
5. On the **Steps** page, configure the job steps. For more details, see Job Properties - Steps Page.
6. On the **Schedules** page, set job schedules. For more details, see Job Properties - Schedules Page.
7. On the **Alerts** page, configure job alerts. For more details, see Job Properties - Alerts Page.
8. On the **Notifications** page, configure job completion notifications. For more details, see Job Properties - Notifications Page.
9. On the **Targets** page, configure the target servers. For more details, see Job Properties - Targets Page.
10. Select **OK** to save the job.

How to Create a Job Using Transact-SQL (T-SQL)

1. In **Object Explorer**, connect to the server.

2. Open a **New Query** window.

3. Copy and paste the following script:

```SQL
USE msdb ;
GO
EXEC dbo.sp_add_job @job_name = N'Weekly Sales Data Backup' ;
GO
EXEC sp_add_jobstep
    @job_name = N'Weekly Sales Data Backup',
    @step_name = N'Set database to read only',
    @subsystem = N'TSQL',
    @command = N'ALTER DATABASE SALES SET READ_ONLY',
    @retry_attempts = 5,
    @retry_interval = 5 ;
GO
EXEC dbo.sp_add_schedule
    @schedule_name = N'RunOnce',
    @freq_type = 1,
    @active_start_time = 233000 ;
GO
EXEC sp_attach_schedule
    @job_name = N'Weekly Sales Data Backup',
    @schedule_name = N'RunOnce';
```

```
GO
EXEC dbo.sp_add_jobserver @job_name = N'Weekly Sales Data Backup';
GO
```

For more details, see:

- sp_add_job (Transact-SQL)
- sp_add_jobstep (Transact-SQL)
- sp_add_schedule (Transact-SQL)
- sp_attach_schedule (Transact-SQL)
- sp_add_jobserver (Transact-SQL)

Use SQL Server Management Objects

To create a SQL Server Agent job using SQL Server Management Objects (SMO):

Call the **Create** method of the **Job** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell. For example code, see Scheduling Automatic Administrative Tasks in SQL Server Agent.

Related content

- Create a Job Step
- Create a Job Category

Feedback

Was this page helpful? Yes No

Create a multiserver environment

Article • 02/13/2025

Applies to: SQL Server

Multiserver administration requires that you set up a master server (MSX) and one or more target servers (TSX). Jobs that are processed on all the target servers are first defined on the master server and then downloaded to the target servers.

> **Important**
>
> Most, but not all, SQL Server Agent features are currently supported on Azure SQL Managed Instance. The Multi Server Administration feature is not supported on Azure SQL Managed Instance.

By default, full Transport Layer Security (TLS), previously known as Secure Sockets Layer (SSL), encryption and certificate validation are enabled for connections between master servers and target servers. For more information, see Set Encryption Options on Target Servers.

If you have a large number of target servers, avoid defining your master server on a production server that has significant performance requirements from other SQL Server functionality, because target server traffic can slow performance on your production server. If you also forward events to a dedicated master server, you can centralize administration on one server. For more information, see Manage Events.

> **Note**
>
> By default, the SQL Server Agent service account is mapped to the default SQL Server Agent service SID (`NT SERVICE\SQLSERVERAGENT`), which is a member of the **sysadmin** fixed server role. The account must also be a member of the `msdb` database role **TargetServersRole** on the master server if multiserver job processing is used. The Master Server Wizard automatically adds the service account to this role as part of the enlistment process.

Considerations for multiserver environments

Consider the following issues when creating a multiserver environment:

- Use the most recent version as the master server. The current and two previous versions are supported.
- Each target server reports to only one master server. You must defect a target server from one master server before you can enlist it into a different one.
- When changing the name of a target server, you must defect it before changing the name and re-enlist it after the change.
- If you want to dismantle a multiserver configuration, you must defect all the target servers from the master server.
- SQL Server Integration Services only supports target servers that are the same version or higher than the master server version.

Related tasks

- Make a Master Server
- Make a Target Server
- Enlist a Target Server to a Master Server
- Defect a Target Server from a Master Server
- Defect Multiple Target Servers from a Master Server
- sp_help_targetserver (Transact-SQL)
- sp_help_targetservergroup (Transact-SQL)

Related content

- Troubleshoot Multiserver Jobs That Use Proxies

Feedback

Was this page helpful? Yes No

Create a PowerShell Script Job Step

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to create and define a SQL Server Agent job step that executes a PowerShell script in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Security

For detailed information, see Implement SQL Server Agent Security.

> *We are listening:* If you find something outdated or incorrect in this article, such as a step or a code example, please tell us. You can click the **This page** button in the **Feedback** section at the bottom of this page. We read every item of feedback about SQL, typically the next day. Thanks.

Using SQL Server Management Studio

To create a PowerShell Script job step

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**, create a new job or right-click an existing job, and then click **Properties**. For more information on creating a job, see Creating Jobs.
3. In the **Job Properties** dialog, click the **Steps** page, and then click **New**.
4. In the **New Job Step** dialog, type a job **Step name**.

5. In the **Type** list, click **PowerShell**.
6. In the **Run as** list, select the proxy account with the credentials that the job will use.
7. In the **Command** box, enter the PowerShell script syntax that will be executed for the job step. Alternately, click **Open** and select a file containing the script syntax. For an example of a PowerShell script, see **Using Transact-SQL** below.
8. Click the **Advanced** page to set the following job step options: what action to take if the job step succeeds or fails, how many times SQL Server Agent should try to execute the job step, and how often retry attempts should be made.

Using Transact-SQL

To create a PowerShell Script job step

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- creates a PowerShell job step that finds the processes
-- that use more than 1000 MB of memory and kills them
USE msdb;
GO
EXEC sp_add_jobstep
    @job_name = N'Weekly Sales Data Backup',
    @step_name = N'Kills all processes that use more than 1000 MB of
memory',
    @subsystem = N'PowerShell',
    @command = N'Get-Process | Where-Object { $_.WS -gt 1000MB } |
Stop-Process',
    @retry_attempts = 5,
    @retry_interval = 5 ;
GO
```

For more information, see sp_add_jobstep (Transact-SQL).

Using SQL Server Management Objects

To create a PowerShell Script job step

Use the **JobStep** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell.

Feedback

Was this page helpful? Yes No

Create a Schedule

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

You can create a schedule for SQL Server Agent jobs in SQL Server by using SQL Server Management Studio, Transact-SQL, or SQL Server Management Objects.

- **Before you begin:**

  Security

- **To create a schedule, using:**

  SQL Server Management Studio

  Transact-SQL

  SQL Server Management Objects

Before You Begin

Security

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To create a schedule

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.

2. Expand **SQL Server Agent**, right-click **Jobs**, and select **Manage Schedules**.

3. In the **Manage Schedules** dialog box, click **New**.
4. In the **Name** box, type a name for the new schedule.
5. If you do not want the schedule to take effect immediately after it has been created, clear the **Enabled** check box.
6. For **Schedule Type**, select one of the following:
    - To start the job when the CPUs reach an idle condition, click **Start whenever the CPUs become idle**.
    - If you want a schedule to run repeatedly, click **Recurring**. To set the recurring schedule, complete the **Frequency**, **Daily Frequency**, and **Duration** groups on the dialog.
    - If you want the schedule to run only one time, click **One time**. To set the **One time** schedule, complete the **One-time occurrence** group on the dialog box.

Using Transact-SQL

To create a schedule

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- creates a schedule named RunOnce.
-- The schedule runs one time, at 23:30 on the day that the schedule is
created.
USE msdb ;
GO

EXEC dbo.sp_add_schedule
    @schedule_name = N'RunOnce',
    @freq_type = 1,
    @active_start_time = 233000 ;

GO
```

For more information, see sp_add_schedule (Transact-SQL).

Using SQL Server Management Objects

To create a schedule

Use the **JobSchedule** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell. For more information, see SQL Server Management Objects (SMO).

Feedback

Was this page helpful? Yes No

Create a SQL Server Agent Master Job

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to create a master Microsoft SQL Server Agent job in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Limitations and Restrictions

Changes to master SQL Server Agent jobs must be propagated to all involved target servers. Because target servers do not initially download a job until those targets are specified, Microsoft recommends that you complete all job steps and job schedules for a particular job before you specify any target servers. Otherwise, you must manual request that the target servers download the modified job again, either by executing the **sp_post_msx_operation** stored procedure or modifying the job using SQL Server Management Studio. For more information, see sp_post_msx_operation (Transact-SQL) or Modify a Job.

Security

Permissions

Distributed jobs that have steps which are associated with a proxy run under the context of the proxy account on the target server. Make sure that the following conditions are met or job steps that are associated with a proxy will not be downloaded from the master server to the target:

- The registry subkey **\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\<*instance_name*>\SQL Server**

Agent\AllowDownloadedJobsToMatchProxyName (REG_DWORD) is set to 1 (true). By default, this subkey is set to 0 (false).

- A proxy account exists on the target server that has the same name as the master server proxy account under which the job step runs.

If job steps that use proxy accounts fail when downloading them from the master server to the target server, you can check the **error_message** column in the **sysdownloadlist** table in the **msdb** database for the following error messages:

- "The job step requires a proxy account, however proxy matching is disabled on the target server." To resolve this error, set the **AllowDownloadedJobsToMatchProxyName** registry subkey to 1.
- "Proxy not found." To resolve this error, make sure a proxy account exists on the target server that has the same name as the master server proxy account under which the job step runs.

Using SQL Server Management Studio

To create a master SQL Server Agent job

1. In the **Object Explorer**, click the plus sign to expand the server where you want to create a SQL Server Agent job.
2. Click the plus sign to expand **SQL Server Agent**.
3. Right-click the **Jobs** folder and select **New Job...**.
4. In the **New Job** dialog box, on the **General** page, modify the general properties of the job. For more information on the available options on this page, see Job Properties - New Job (General Page)
5. On the **Steps** page, organize the job steps. For more information on the available options on this page, see Job Properties - New Job (Steps Page)
6. On the **Schedules** page, organize schedules for the job. For more information on the available options on this page, see Job Properties - New Job (Schedules Page)
7. On the **Alerts** page, organize the alerts for the job. For more information on the available options on this page, see Job Properties - New Job (Alerts Page)
8. On the **Notifications** page, set actions for Microsoft SQL Server Agent to perform when the job completes. For more information on the available options on this

page, see Job Properties - New Job (Notifications Page).

9. On the **Targets** page, manage the target servers for the job. For more information on the available options on this page, see Job Properties - New Job (Targets Page).

10. When finished, click **OK**.

Using Transact-SQL

To create a master SQL Server Agent job

1. In **Object Explorer**, connect to an instance of Database Engine.

2. On the Standard bar, click **New Query**.

3. Copy and paste the following example into the query window and click **Execute**.

```
USE msdb ;
GO
-- Adds a new job executed by the SQLServerAgent service called 'Weekly
Sales Data Backup'
EXEC dbo.sp_add_job
    @job_name = N'Weekly Sales Data Backup' ;
GO
-- Adds a step (operation) to the 'Weekly Sales Data Backup' job.
EXEC sp_add_jobstep
    @job_name = N'Weekly Sales Data Backup',
    @step_name = N'Set database to read only',
    @subsystem = N'TSQL',
    @command = N'ALTER DATABASE SALES SET READ_ONLY',
    @retry_attempts = 5,
    @retry_interval = 5 ;
GO
-- Creates a schedule called RunOnce
EXEC dbo.sp_add_schedule
    @schedule_name = N'RunOnce',
    @freq_type = 1,
    @active_start_time = 233000 ;
USE msdb ;
GO
-- Sets the 'RunOnce' schedule to the "Weekly Sales Data Backup' Job
EXEC sp_attach_schedule
   @job_name = N'Weekly Sales Data Backup',
   @schedule_name = N'RunOnce';
GO
-- assigns the multiserver job Weekly Sales Backups to the server
SEATTLE2
-- assumes that SEATTLE2 is registered as a target server for the
```

```
current instance.
EXEC dbo.sp_add_jobserver
    @job_name = N'Weekly Sales Data Backups',
    @server_name = N'SEATTLE2' ;
GO
```

For more information, see:

- sp_add_job (Transact-SQL)
- sp_add_jobstep (Transact-SQL)
- sp_add_schedule (Transact-SQL)
- sp_attach_schedule (Transact-SQL)
- sp_add_jobserver (Transact-SQL)

Feedback

Was this page helpful? Yes No

Create a SQL Server Agent proxy

Article • 02/13/2025

Applies to: SQL Server

This article describes how to create a SQL Server Agent proxy in SQL Server by using SQL Server Management Studio or Transact-SQL.

A SQL Server Agent proxy account defines a security context in which a job step can run. Each proxy corresponds to a security credential. To set permissions for a particular job step, create a proxy that has the required permissions for a SQL Server Agent subsystem, and then assign that proxy to the job step.

On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server or SQL Agent job limitations in SQL Managed Instance for details.

Limitations

You must create a credential before you create a proxy if one isn't already available.

SQL Server Agent proxies use credentials to store information about Windows user accounts. The user specified in the credential must have "Access this computer from the network" permission (`SeNetworkLogonRight`) on the computer on which SQL Server is running.

SQL Server Agent checks subsystem access for a proxy and gives access to the proxy each time the job step runs. If the proxy no longer has access to the subsystem, the job step fails. Otherwise, SQL Server Agent impersonates the user that is specified in the proxy and runs the job step. For a list of proxy subsystems, see sp_grant_proxy_to_subsystem.

Creation of a proxy doesn't change the permissions for the user that is specified in the credential for the proxy. For example, you can create a proxy for a user that doesn't have permission to connect to an instance of SQL Server. In this case, job steps that use that proxy are unable to connect to SQL Server.

If the login for the user has access to the proxy, or the user belongs to any role with access to the proxy, the user can use the proxy in a job step.

Permissions

Only members of the **sysadmin** fixed server role have permission to create, modify, or delete proxy accounts. Users who aren't members of the **sysadmin** fixed server role must be added to one of the following SQL Server Agent fixed database roles in the `msdb` database to use proxies: **SQLAgentUserRole**, **SQLAgentReaderRole**, or **SQLAgentOperatorRole**.

Requires `ALTER ANY CREDENTIAL` permission if creating a credential in addition to the proxy.

Use SQL Server Management Studio (SSMS)

1. In **Object Explorer**, select the plus sign to expand the server where you want to create a proxy on SQL Server Agent.
2. Select the plus sign to expand **SQL Server Agent**.
3. Right-click the **Proxies** folder and select **New Proxy**.
4. On the **New Proxy Account** dialog box, on the **General** page, enter the name of the proxy account in the **Proxy name** box.
5. In the **Credential name** box, enter the name of the security credential that the proxy account will use.
6. In the **Description** box, enter a description for the proxy account
7. Under **Active to the following subsystems**, select the appropriate subsystem or subsystems for this proxy.
8. On the **Principals** page, add or remove logins or roles to grant or remove access to the proxy account.
9. When finished, select **OK**.

Use Transact-SQL

The following script creates a credential called `CatalogApplicationCredential`, creates proxy `Catalog application proxy` and assigns the credential `CatalogApplicationCredential` to it, and grants the proxy access to the ActiveX Scripting subsystem.

1. Create credential `CatalogApplicationCredential`.

```
```

```SQL
USE msdb;
GO

CREATE CREDENTIAL CatalogApplicationCredential
    WITH IDENTITY = 'REDMOND/TestUser', SECRET = 'G3$1o)lkJ8HNd!';
GO
```

2. Create proxy `Catalog application proxy` and assign the credential `CatalogApplicationCredential` to it.

```SQL
EXECUTE dbo.sp_add_proxy
    @proxy_name = 'Catalog application proxy',
    @enabled = 1,
    @description = 'Maintenance tasks on catalog application.',
    @credential_name = 'CatalogApplicationCredential';
GO
```

3. Grant the proxy `Catalog application proxy` access to the ActiveX Scripting subsystem.

```SQL
EXECUTE dbo.sp_grant_proxy_to_subsystem
    @proxy_name = N'Catalog application proxy',
    @subsystem_id = 2;
GO
```

Related content

- CREATE CREDENTIAL (Transact-SQL)
- sp_add_proxy (Transact-SQL)
- sp_grant_proxy_to_subsystem (Transact-SQL)

Feedback

Was this page helpful? Yes No

Create a Transact-SQL Job Step

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to create a Microsoft SQL Server Agent job step that executes Transact-SQL scripts in SQL Server by using SQL Server Management Studio, Transact-SQL, or SQL Server Management Objects.

These job step scripts may call stored procedures and extended stored procedures. A single Transact-SQL job step can contain multiple batches and embedded GO commands. For more information on creating a job, see Creating Jobs.

Before You Begin

Security

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To create a Transact-SQL job step

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**, create a new job or right-click an existing job, and then click **Properties**.
3. In the **Job Properties** dialog, click the **Steps** page, and then click **New**.
4. In the **New Job Step** dialog, type a job **Step name**.
5. In the **Type** list, click **Transact-SQL Script (TSQL)**.

6. In the **Command** box, type the Transact-SQL command batches, or click **Open** to select a Transact-SQL file to use as the command.

7. Click **Parse** to check your syntax.

8. The message "Parse succeeded" is displayed when your syntax is correct. If an error is found, correct the syntax before continuing.

9. Click the **Advanced** page to set job step options, such as: what action to take if the job step succeeds or fails, how many times SQL Server Agent should try to execute the job step, and the file or table where SQL Server Agent can write the job step output. Only members of the **sysadmin** fixed server role can write job step output to an operating system file. All SQL Server Agent users can log output to a table.

10. If you are a member of the **sysadmin** fixed server role and you want to run this job step as a different SQL login, select the SQL login from the **Run as user** list.

Using Transact-SQL

To create a Transact-SQL job step

1. In **Object Explorer**, connect to an instance of Database Engine.

2. On the Standard bar, click **New Query**.

3. Copy and paste the following example into the query window and click **Execute**.

```
-- creates a job step that uses Transact-SQL
USE msdb;
GO
EXEC sp_add_jobstep
    @job_name = N'Weekly Sales Data Backup',
    @step_name = N'Set database to read only',
    @subsystem = N'TSQL',
    @command = N'ALTER DATABASE SALES SET READ_ONLY',
    @retry_attempts = 5,
    @retry_interval = 5 ;
GO
```

For more information, see sp_add_jobstep (Transact-SQL).

Using SQL Server Management Objects

To create a Transact-SQL job step

Use the **JobStep** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell.

Feedback

Was this page helpful? Yes No

Create a User-Defined Event

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

You can create user-defined events if you want to monitor events other than events that are predefined by SQL Server. You can also assign a severity level to each user-defined event.

> **Note**
>
> When using SQL Server Management Studio, select the **Write to Windows application event log** option for each user-defined event message, to ensure that the messages are logged. By default, user-defined messages of severity lower than 19 are not sent to the Microsoft Windows application log when they occur. User-defined messages of severity lower than 19 therefore do not trigger SQL Server Agent alerts.

User-defined events must have a unique message number. Message numbers for a user-defined event must be greater than 50,000. You can define messages for the event in multiple languages. However, an **En-US** error message must exist before messages in other languages can be added.

If you administer a multiple-language SQL Server environment, create user-defined messages in each of the languages that are supported. For example, if you are creating a new event message to be used on both an English and a German server, use the same message number and severity for both, but assign a different language to each.

The following tasks provide information about how to create user-defined events and alerts that respond to them:

To create an alert based on a message number

- SQL Server Management Studio

- Transact-SQL

To create an alert based on severity levels

- SQL Server Management Studio
- Transact-SQL

To define the response to an alert

- SQL Server Management Studio
- Transact-SQL

To create a user-defined event error message

- Transact-SQL

To modify a user-defined event error message

- Transact-SQL

To delete a user-defined event error message

- Transact-SQL

To disable or reactivate an alert

- SQL Server Management Studio
- Transact-SQL

See Also

sp_update_alert (Transact-SQL)

Feedback

Was this page helpful? Yes No

Create a WMI event alert

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This article describes how to a SQL Server Agent alert that is raised when a specific SQL Server event occurs that is monitored by the WMI Provider for Server Events in SQL Server by using SQL Server Management Studio or Transact-SQL.

For information about the using the WMI Provider to monitor SQL Server events, see WMI Provider for Server Events Classes and Properties. For information about the permissions necessary to receive WMI event alert notifications, see Select an Account for the SQL Server Agent Service. For more information about WQL, see Using WQL with the WMI Provider for Server Events.

Limitations and restrictions

- SQL Server Management Studio provides an easy, graphical way to manage the entire alerting system and is the recommended way to configure an alert infrastructure.
- Events generated with `xp_logevent` occur in the `master` database. Therefore, `xp_logevent` does not trigger an alert unless the `@database_name` for the alert is `'master'` or NULL.
- Only WMI namespaces on the computer that runs SQL Server Agent are supported.

Permissions

By default, only members of the **sysadmin** fixed server role can execute `sp_add_alert`.

Use SQL Server Management Studio

1. In **Object Explorer,** select the plus sign to expand the server where you want to create a WMI event alert.
2. Select the plus sign to expand **SQL Server Agent**.
3. Right-click **Alerts** and select **New Alert**.
4. In the **New Alert** dialog box, in the **Name** box, enter a name for this alert.
5. Check the **Enable** check box to enable the alert to run. By default, **Enable** is checked.
6. In the **Type** list, select **WMI event alert**.
7. Under **WMI event alert definition**, in the **Namespace** box, specify the WMI namespace for the WMI Query Language (WQL) statement that identifies which WMI event will trigger this alert.
8. In the **Query** box, specify the WQL statement that identifies the event that this alert responds to.
9. Select **OK**.

Use Transact-SQL

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, select **New Query**.
3. Copy and paste the following example into the query window and select **Execute**.

```SQL
USE msdb;
GO

EXEC dbo.sp_add_alert @name = N'Test Alert 2',
    @message_id = 54001,
    @notification_message = N'Error 54001 has occurred on the
Sales.SalesOrderDetail table on the AdventureWorks2022 database.',
    @wmi_namespace = '\.\root\Microsoft\SqlServer\ServerEvents',
    @wmi_query = N'SELECT * FROM ALTER_TABLE
WHERE DatabaseName = ''AdventureWorks2022'' AND SchemaName = ''Sales''
AND ObjectType=''Table'' AND ObjectName = ''SalesOrderDetail''';
GO
```

Next steps

- sp_add_alert (Transact-SQL)

Feedback

Was this page helpful? Yes No

Create an ActiveX script job step

Article • 02/13/2025

Applies to: SQL Server

ActiveX subsystem is discontinued beginning with SQL Server 2016. Convert any existing job steps that use ActiveX Script to a PowerShell script job step. Use PowerShell for any future development.

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance from SQL Server for details.

This topic describes how to create and define a Microsoft SQL Server Agent job step in SQL Server 2014 and prior that executes an ActiveX script by using SQL Server Management Studio, Transact-SQL, or SQL Server Management Objects.

Before you begin

Limitations and Restrictions

This feature will be removed in a future version of SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature.

Security

For detailed information, see Implement SQL Server Agent Security.

Use SQL Server Management Studio

To create an ActiveX Script job step

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.

2. Expand **SQL Server Agent**, create a new job or right-click an existing job, and then click **Properties**. For more information on creating a job, see Creating Jobs.

3. In the **Job Properties** dialog, click the **Steps** page, and then click **New**.

4. In the **New Job Step** dialog, type a job **Step name**.

5. In the **Type** list, click **ActiveX Script**.

6. In the **Run as** list, select the proxy account with the credentials that the job will use.

7. Select the **Language** in which the script was written. Alternatively, click **Other** and then enter the name of the Microsoft ActiveX scripting language in which the script will be written.

8. In the **Command** box, enter the script syntax that will be executed for the job step. Alternately, click **Open** and select a file containing the script syntax.

9. Click the **Advanced** page to set the following job step options: what action to take if the job step succeeds or fails, how many times SQL Server Agent should try to execute the job step, and how often retry attempts should be made.

Using Transact-SQL

To create an ActiveX script job step

1. In **Object Explorer**, connect to an instance of Database Engine.

2. On the Standard bar, click **New Query**.

3. Copy and paste the following example into the query window and click **Execute**.

```
-- create an ActiveX Script job step written in VBScript that creates a
restore point
USE msdb;
GO
EXEC sp_add_jobstep
    @job_name = N'Weekly Sales Data Backup',
    @step_name = N'Create a restore point',
    @subsystem = N'ACTIVESCRIPTING',
    @command = N'Const RESTORE_POINT = 20

strComputer = "."
Set objWMIService = GetObject("winmgmts:" _
    & "{impersonationLevel=impersonate}!\\" & strComputer &
```

```
"\root\default")

Set objItem = objWMIService.Get("SystemRestore")
errResults = objItem.Restore(RESTORE_POINT)',
    @retry_attempts = 5,
    @retry_interval = 5 ;
GO
```

For more information, see sp_add_jobstep (Transact-SQL).

Using SQL Server Management Objects

To create an ActiveX Script job step

Use the **JobStep** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell.

Feedback

Was this page helpful? Yes No

Create an Alert Using an Error Number

Article • 02/13/2025

Applies to: SQL Server

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance from SQL Server for details.

This topic describes how to create a Microsoft SQL Server Agent alert occurs in SQL Server that will be raised when an error of a specific number occurs by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Limitations and Restrictions

- SQL Server Management Studio provides an easy, graphical way to manage the entire alerting system and is the recommended way to configure an alert infrastructure.
- Events generated with **xp_logevent** occur in the master database. Therefore, **xp_logevent** does not trigger an alert unless the **@database_name** for the alert is '**master**' or NULL.

Security

Permissions

By default, only members of the **sysadmin** fixed server role can execute **sp_add_alert**.

Using SQL Server Management Studio

To create an alert using an error number

1. In **Object Explorer,** click the plus sign to expand the server where you want to create an alert using an error number.
2. Click the plus sign to expand **SQL Server Agent**.
3. Right-click **Alerts** and select **New Alert**.
4. In the **New Alert** dialog box, in the **Name** box, enter a name for this alert.
5. Check the **Enable** check box to enable the alert to run. By default, **Enable** is checked.
6. In the **Type** list, select **SQL Server event alert**.
7. Under **Event alert definition**, in the **Database name** list, select a database to restrict the alert to a specific database.
8. Under **Alerts will be raised based on**, click **Error number**, and then type a valid error number for the alert. Alternately, click **Severity** and then select the specific severity that will raise the alert.
9. Check the box corresponding to **Raise alert when message contains** check box to restrict the alert to a particular character sequence, and then enter a keyword or character string for the **Message text**. The maximum number of characters is 100.
10. Click **OK**.

Using Transact-SQL

To create an alert using an error number

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- adds an alert (Test Alert) that runs the Back up
-- the AdventureWorks2022 Database job when fired
-- assumes that the message 55001 and the Back up
-- the AdventureWorks2022 Database job already exist.
USE msdb ;
GO
```

```
EXEC dbo.sp_add_alert
    @name = N'Test Alert',
    @message_id = 55001,
   @severity = 0,
   @notification_message = N'Error 55001 has occurred. The DB will be
backed up...',
   @job_name = N'Back up the AdventureWorks2022 Database' ;
GO
```

For more information, see sp_add_alert (Transact-SQL).

Feedback

Was this page helpful? Yes No

Create an Alert Using Severity Level

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to create a Microsoft SQL Server Agent alert that is raised when an event of a specific severity level occurs in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Limitations and Restrictions

- SQL Server Management Studio provides an easy, graphical way to manage the entire alerting system and is the recommended way to configure an alert infrastructure.
- Events generated with **xp_logevent** occur in the master database. Therefore, **xp_logevent** does not trigger an alert unless the **@database_name** for the alert is '**master**' or NULL.
- Severity levels from 19 through 25 send a SQL Server message to the Microsoft Windows application log and trigger an alert. Events with severity levels less than 19 will trigger alerts only if you have used **sp_altermessage**, RAISERROR WITH LOG, or **xp_logevent** to force them to be written to the Windows application log.

Security

Permissions

By default, only members of the **sysadmin** fixed server role can execute **sp_add_alert**.

Using SQL Server Management Studio

To create an alert using severity level

1. In **Object Explorer,** click the plus sign to expand the server where you want to create an alert using severity level.
2. Click the plus sign to expand **SQL Server Agent**.
3. Right-click **Alerts** and select **New Alert**.
4. In the **New Alert** dialog box, in the **Name** box, enter a name for this alert.
5. In the **Type** list, select **SQL Server event alert**.
6. Under **Event alert definition**, in the **Database name** list, select a database to restrict the alert to a specific database.
7. Under **Alerts will be raised based on**, click **Severity** and then select the specific severity that will raise the alert.
8. Check the box corresponding to **Raise alert when message contains** check box to restrict the alert to a particular character sequence, and then enter a keyword or character string for the **Message text**. The maximum number of characters is 100.
9. Click **OK**.

Using Transact-SQL

To create an alert using severity level

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- Adds an alert (Test Alert) that notifies the
-- Alert Operator via email when an error with a
-- severity of 23 is detected.

-- Assumes that the Alert Operator already exists
-- and that database mail is configured.
```

```
USE msdb ;
GO

EXEC dbo.sp_add_alert @name=N'Test Alert',
  @message_id = 0,
  @severity = 23,
  @enabled = 1,
  @include_event_description_in = 1
;
GO

EXEC dbo.sp_add_notification @alert_name=N'Test Alert',
  @operator_name=N'Alert Operator',
  @notification_method=1
;
GO
```

For more information, see sp_add_alert (Transact-SQL).

Feedback

Was this page helpful? Yes No

Create an Analysis Services Job Step

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance from SQL Server for details.

This topic describes how to create and define SQL Server Agent job steps in SQL Server that execute SQL Server Analysis Services commands and queries by using SQL Server Management Studio, Transact-SQL or SQL Server Management Objects.

- **Before you begin:**

  Limitations and Restrictions

  Security

- **To create a SQL Server job steps using Analysis Services commands and/or queries, with:**

  SQL Server Management Studio

  Transact-SQL

  SQL Server Management Objects

Before You Begin

Limitations and Restrictions

- If the job step uses an Analysis Services command, the command statement must be an XML for Analysis Services **Execute** method. The statement may not contain a complete Simple Object Access Protocol (SOAP) envelope or an XML for Analysis **Discover** method. While SQL Server Management Studio supports complete SOAP envelopes and the **Discover** method, SQL Server Agent job steps do not. For more information about XML for Analysis Services, see XML for Analysis Overview (XMLA).

- If the job step uses an Analysis Services query, the query statement must be a multidimensional expressions (MDX) query. For more information about MDX, see MDX Statement Fundamentals (MDX).

Security

Permissions

- To run a job step that uses the Analysis Services subsystem, a user must be a member of the **sysadmin** fixed server role or have access to a valid proxy account defined to use this subsystem. In addition, the SQL Server Agent service account or the proxy must be an Analysis Services administrator and a valid Windows domain account.
- Only members of the **sysadmin** fixed server role can write job step output to a file. If the job step is run by users who are members of the **SQLAgentUserRole** database role in the **msdb** database, then the output can be written only to a table. SQL Server Agent writes job step output to the **sysjobstepslog** table in the **msdb** database.
- For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To create an Analysis Services command job step

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**, create a new job or right-click an existing job, and then click **Properties**. For more information on creating a job, see Create Jobs.
3. In the **Job Properties** dialog box, click the **Steps** page, and then click **New**.
4. In the **New Job Step** dialog box, type a job **Step name**.
5. In the **Type** list, click **SQL Server Analysis Services Command**.
6. In the **Run as** list, select a proxy that has been defined to use the Analysis Services Command subsystem. A user who is a member of the **sysadmin** fixed server role can also select **SQL Agent Service Account** to run this job step.

7. Select the **Server** where the job step will run, or type the server name.
8. In the **Command** box, type the statement to execute, or click **Open** to select a statement.
9. Click the **Advanced** page to define options for this job step, such as what action SQL Server Agent should take if the job step succeeds or fails, how many times the job step should be attempted, and where the job step output should be written.

To create an Analysis Services query job step

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**, create a new job or right-click an existing job, and then click **Properties**. For more information on creating a job, see Create Jobs.
3. In the **Job Properties** dialog, click the **Steps** page, and then click **New**.
4. In the **New Job Step** dialog, type a job **Step name**.
5. In the **Type** list, click **SQL Server Analysis Services Query**.
6. In the **Run as** list, select a proxy that has been defined to use the Analysis Services Query subsystem. A user who is a member of the **sysadmin** fixed server role can also select **SQL Agent Service Account** to run this job step.
7. Select the **Server** and the **Database** where the job step will run, or type the server or database name.
8. In the **Command** box, type the statement to execute, or click **Open** to select a statement.
9. Click the **Advanced** page to define options for this job step, such as what action SQL Server Agent should take if the job step succeeds or fails, how many times the job step should be attempted, and where the job step output should be written.

Using Transact-SQL

To create an Analysis Services command job step

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.

3. Copy and paste the following example into the query window and click **Execute**.

```
-- Creates a job step that uses XMLA to create a relational data source
that
-- references the AdventureWorks2022 Microsoft SQL Server database.
USE msdb;
GO
EXEC sp_add_jobstep
    @job_name = N'Weekly Sales Data Backup',
    @step_name =
        N'Create a relational data source that references the
AdventureWorks2022 Microsoft SQL Server database',
    @subsystem = N'ANALYSISCOMMAND',
    @command =
        N' <Create
xmlns="http://schemas.microsoft.com/analysisservices/2003/engine">
    <ParentObject>
        <DatabaseID>AdventureWorks2022</DatabaseID>
    </ParentObject>
    <ObjectDefinition>
        <DataSource xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:type="RelationalDataSource">
            <ID>AdventureWorks2022</ID>
            <Name>Adventure Works 2022</Name>
            <ConnectionString>Data Source=localhost;Initial
Catalog=AdventureWorks2022;Integrated Security=True</ConnectionString>
            <ImpersonationInfo>

<ImpersonationMode>ImpersonateServiceAccount</ImpersonationMode>
            </ImpersonationInfo>
            <ManagedProvider>System.Data.SqlClient</ManagedProvider>
            <Timeout>PT0S</Timeout>
        </DataSource>
    </ObjectDefinition>
</Create>', ;
GO
```

For more information, see sp_add_jobstep (Transact-SQL).

To create an Analysis Services query job step

1. In **Object Explorer**, connect to an instance of Database Engine.

2. On the Standard bar, click **New Query**.

3. Copy and paste the following example into the query window and click **Execute**.

```
-- Creates a job step that uses MDX to return data
USE msdb;
GO
EXEC sp_add_jobstep
    @job_name = N'Weekly Sales Data Backup',
    @step_name = N'Returns the Internet sales amount by state',
    @subsystem = N'ANALYSISQUERY',
    @command = N' SELECT
   [Measures].[Internet Sales Amount] ON COLUMNS,
   [Customer].[State-Province].Members ON ROWS
FROM [AdventureWorks2022]',
    @retry_attempts = 5,
    @retry_interval = 5 ;
GO
```

For more information, see sp_add_jobstep (Transact-SQL).

Using SQL Server Management Objects

To create a PowerShell Script job step

Use the **JobStep** class by using a programming language that you choose, such as XMLA or MDX. For more information, see SQL Server Management Objects (SMO).

Feedback

Was this page helpful? Yes No

Create an Operator

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to configure a user to receive notifications about Microsoft SQL Server Agent jobs in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Limitations and Restrictions

- The Pager and **net send** options will be removed from SQL Server Agent in a future version of Microsoft SQL Server. Avoid using these features in new development work, and plan to modify applications that currently use these features.
- Note that SQL Server Agent must be configured to use Database Mail to send e-mail and pager notifications to operators. For more information, see Assign Alerts to an Operator.
- SQL Server Management Studio provides an easy, graphical way to manage jobs, and is the recommended way to create and manage the job infrastructure.

Security

Permissions

Only members of the **sysadmin** fixed server role can create operators.

Using SQL Server Management Studio

To create an operator

1. In **Object Explorer**, click the plus sign to expand the server where you want to create a SQL Server Agent operator.

2. Click the plus sign to expand **SQL Server Agent**.

3. Right-click the **Operators** folder and select **New Operator**.

   The following options are available on the **General** page of the **New Operator** dialog box:

   Name
   Change the name of the operator.

   Enabled
   Enable the operator. When not enabled, no notifications are sent to the operator.

   E-mail name
   Specifies the e-mail address for the operator.

   Net send address
   Specify the address to use for **net send**.

   Pager e-mail name
   Specifies the e-mail address to use for the operator's pager.

   Pager on duty schedule
   Sets the times at which the pager is active.

   Monday - Sunday
   Select the days that the pager is active.

   Workday begin
   Select the time of day after which SQL Server Agent sends messages to the pager.

   Workday end
   Select the time of day after which SQL Server Agent no longer sends messages to the pager.

   The following options are available on the **Notifications** page of the **New Operator** dialog box:

   Alerts
   View the alerts in the instance.

Jobs
View the jobs in the instance.

Alert list
Lists the alerts in the instance.

Job list
Lists the jobs in the instance.

E-mail
Notify this operator using e-mail.

Pager
Notify this operator by sending e-mail to the pager address.

Net send
Notify this operator using **net send**.

4. When finished creating the new operator, click **OK**.

Using Transact-SQL

To create an operator

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- sets up the operator information for user 'danwi.'
-- The operator is enabled.
-- SQL Server Agent sends notifications by pager
-- from Monday through Friday from 8 A.M. to 5 P.M.
USE msdb ;
GO

EXEC dbo.sp_add_operator
    @name = N'Dan Wilson',
    @enabled = 1,
    @email_address = N'danwi',
    @pager_address = N'5551290AW@pager.Adventure-Works.com',
    @weekday_pager_start_time = 080000,
    @weekday_pager_end_time = 170000,
```

```
    @pager_days = 62 ;
GO
```

For more information, see sp_add_operator (Transact-SQL).

Feedback

Was this page helpful? Yes No

Create and Attach Schedules to Jobs

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Scheduling SQL Server Agent jobs means defining the condition or conditions that cause the job to begin running without user interaction. You can schedule a job to run automatically by creating a new schedule for the job, or by attaching an existing schedule to the job.

There are two ways to create a schedule:

- Create the schedule while you are creating a job.
- Create the schedule in Object Explorer.

After a schedule has been created, you can attach that schedule to multiple jobs, even if the schedule was created for a specific job. You can also detach schedules from jobs.

> **Warning**
>
> Updating a schedule will impact all the jobs attached to the schedule. If you want to control the schedule of individual jobs, be sure to create a separate schedule object for each job.

A schedule can be based upon time or an event. For example, you can schedule a job to run at the following times:

- Whenever SQL Server Agent starts.
- Whenever CPU utilization of the computer is at a level you have defined as idle.
- One time, at a specific date and time.
- On a recurring schedule.

As an alternative to job schedules, you can also create an alert that responds to an event by running a job.

> **Note**
>
> Only one instance of the job can be run at a time. If you try to run a job manually while it is running as scheduled, SQL Server Agent refuses the request.

To prevent a scheduled job from running, you must do one of the following:

- Disable the schedule.
- Disable the job.
- Detach the schedule from the job.
- Stop the SQL Server Agent service.
- Delete the schedule.

If the schedule is not enabled, the job can still run in response to an alert or when a user runs the job manually. When a job schedule is not enabled, the schedule is not enabled for any job that uses the schedule.

You must explicitly re-enable a schedule that has been disabled. Editing the schedule does not automatically re-enable the schedule.

Scheduling Start Dates

The start date of a schedule must be greater than or equal to 19900101.

When you are attaching a schedule to a job, you should review the start date that the schedule uses to run the job for the first time. The start date depends upon the day and time when you attach the schedule to the job. For example, you create a schedule that runs every other Monday at 8:00 A.M. If you create a job at 10:00 A.M. on Monday, March 3, 2008, the schedule start date is Monday, March 17, 2008. If you create another job on Tuesday, March 4, 2008, the schedule start date is Monday, March 10, 2008.

You can change the schedule start date after you attach the schedule to a job.

CPU Idle Schedules

To maximize CPU resources, you can define a CPU idle condition for SQL Server Agent. SQL Server Agent uses the CPU idle condition setting to determine the best time to run jobs. For example, you can schedule a job to rebuild indexes during CPU idle time and slow production periods.

Before you define jobs to run during CPU idle time, determine the load on the CPU during normal processing. To do this, use SQL Server Profiler or Performance Monitor to monitor server traffic and collect statistics. You can then use the information you gather to set the CPU idle time percentage and duration.

Define the CPU idle condition as a percentage below which CPU usage must remain for a specified time. Next, set the amount of time. When the CPU usage is below the specified percentage for the specified amount of time, SQL Server Agent starts all jobs that have a CPU idle time schedule. For more information on using SQL Server Profiler or Performance Monitor to monitor CPU usage, see Monitoring CPU Usage.

Related Tasks

Expand table

Description	Topic
Describes how to create a schedule for a SQL Server Agent job.	Create a Schedule
Describes how to schedule a SQL Server Agent job.	Schedule a Job
Explains how to define the CPU idle condition for your server.	Set CPU Idle Time and Duration (SQL Server Management Studio)

See Also

sp_help_jobschedule
sysjobschedules

Feedback

Was this page helpful? Yes No

Create Jobs

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

A job is a specified series of operations performed sequentially by SQL Server Agent. A job can perform a wide range of activities, including running Transact-SQL scripts, command prompt applications, Microsoft ActiveX scripts, Integration Services packages, Analysis Services commands and queries, or Replication tasks. Jobs can run repetitive or schedulable tasks, and they can automatically notify users of job status by generating alerts, thereby greatly simplifying SQL Server administration.

To create a job, a user must be a member of one of the SQL Server Agent fixed database roles or the **sysadmin** fixed server role. A job can be edited only by its owner or members of the **sysadmin** role. Members of the **sysadmin** role can assign job ownership to other users, and they can run any job, regardless of the job owner. For more information about the SQL Server Agent fixed database roles, see SQL Server Agent Fixed Database Roles.

Jobs can be written to run on the local instance of SQL Server or on multiple instances across an enterprise. To run jobs on multiple servers, you must set up at least one master server and one or more target servers. For more information about master and target servers, see Automated Administration Across an Enterprise

SQL Server Agent records job and job step information in the job history.

Related Tasks

Expand table

Description	Topic
Describes how to create a SQL Server Agent job.	Create a Job

Description	Topic
Describes how to reassign ownership of SQL Server Agent jobs to another user.	Give Others Ownership of a Job
Describes how to set up the SQL Server Agent job history log.	Set Up the Job History Log

See Also

Manage Job Steps
Automated Administration Across an Enterprise
Create and Attach Schedules to Jobs
Run Jobs
View or Modify Jobs

Feedback

Was this page helpful? Yes No

Defect a Target Server from a Master Server

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to defect a target server from a master server in SQL Server by using SQL Server Management Studio, Transact-SQL, or SQL Server Management Objects (SMO). Run this procedure from the target server.

Before You Begin

Security

Permissions

To execute this stored procedure, a user must be a member of the **sysadmin** fixed server role.

Using SQL Server Management Studio

To defect a target server from a master server

1. In **Object Explorer**, expand a server that is configured as a target server.
2. Right-click **SQL Server Agent**, point to **Multi Server Administration**, and then click **Defect**.
3. Click **Yes** to confirm that you want to defect this target server from a master server.

Using Transact-SQL

To defect a target server from a master server

1. Connect to the Database Engine.
2. From the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
sp_msx_defect ;
```

For more information, see sp_msx_defect (Transact-SQL).

Using SQL Server Management Objects (SMO)

Use the **MsxDefect Method**.

See Also

Create a Multiserver Environment
Automated Administration Across an Enterprise
Defect Multiple Target Servers from a Master Server

Feedback

Was this page helpful? Yes No

Defect Multiple Target Servers from a Master Server

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance from SQL Server for details.

This topic describes how to defect multiple target servers from a multiserver administration configuration in SQL Server by using SQL Server Management Studio. Run this procedure from the master server.

Using SQL Server Management Studio

To defect multiple target servers from a master server

1. In **Object Explorer**, expand a server that is configured as a master server.
2. Right-click **SQL Server Agent**, point to **Multi Server Administration**, and then click **Manage Target Servers**.
3. Click **Post Instructions**, and then in the **Instruction type** list, select **Defect**.
4. Under **Recipients**, do one of the following:
   - Click **All target servers** to defect all target servers of this master server. Use this option if you want to completely uninstall the current multiserver administration configuration.
   - Click **These target servers**, and then click the corresponding **Select** box, to defect some, but not all, target servers of this master server.

See Also

Create a Multiserver Environment
Automated Administration Across an Enterprise
Defect a Target Server from a Master Server

Feedback

Was this page helpful? Yes No

Define the Response to an Alert

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to define how Microsoft SQL Server responds to SQL Server Agent alerts in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Limitations and Restrictions

- The Pager and **net send** options will be removed from SQL Server Agent in a future version of Microsoft SQL Server. Avoid using these features in new development work, and plan to modify applications that currently use these features.
- Note that SQL Server Agent must be configured to use Database Mail to send e-mail and pager notifications to operators. For more information, see Assign Alerts to an Operator.
- SQL Server Management Studio provides an easy, graphical way to manage jobs, and is the recommended way to create and manage the job infrastructure.

Security

Permissions

Only members of the **sysadmin** fixed server role can define the response to an alert.

Using SQL Server Management Studio

To define the response to an alert

1. In **Object Explorer**, click the plus sign to expand the server that contains the alert on which you want to define a response.
2. Click the plus sign to expand **SQL Server Agent**.
3. Click the plus sign to expand the **Alerts** folder.
4. Right-click the alert on which you want to define a response and select **Properties**.
5. In the *alert_name***alert properties** dialog box, under **Select a page**, select **Response**.
6. Select the **Execute job** check box and, from the list below the **Execute job** check box, select a job to execute when the alert occurs. You can create a new job by clicking **New Job**. You can view more information about the job by clicking **View Job**. For more information about the available options in the **New Job** and **Job Properties***job_name* dialog boxes, see Create a Job and View a Job.
7. Select the **Notify Operators** check box if you want to notify operators when the alert is activated. In the **Operator list**, select one or more of the following methods for notifying the operator or operators: **E-mail**, **Pager**, or **Net Send**. You can create a new operator by clicking **New Operator**. You can view more information about an operator by clicking **View Operator**. For more information about the available options in the **New Operator** and **View Operator Properties** dialog boxes, see Create an Operator and View Information About an Operator.
8. When finished, click **OK**.

Using Transact-SQL

To define the response to an alert

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- adds an e-mail notification for Test Alert.
-- assumes that Test Alert already exists and that
-- François Ajenstat is a valid operator name
USE msdb ;
GO
```

```
EXEC dbo.sp_add_notification
 @alert_name = N'Test Alert',
 @operator_name = N'François Ajenstat',
 @notification_method = 1 ;
GO
```

For more information, see sp_add_notification (Transact-SQL).

Feedback

Was this page helpful? Yes No

Define Transact-SQL Job Step Options

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to define options for Microsoft SQL Server Agent Transact-SQL job steps in SQL Server by using SQL Server Management Studio or SQL Server Management Objects.

Before You Begin

Security

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To define Transact-SQL job step options

1. In **Object Explorer**, expand **SQL Server Agent**, expand **Jobs**, right-click the job you want to edit, and then click **Properties**.
2. Click the **Steps** page, click a job step, and then click **Edit**.
3. On the **Job Step Properties** dialog, confirm that the job type is **Transact-SQL script (TSQL)**, and then select the **Advanced** page.
4. Specify an action to take if the job is successful by selecting from the **On success action** list.
5. Specify a number of retry attempts by entering a number from 0 to 9999 into the **Retry attempts** box.

6. Specify a retry interval by entering a number of minutes from 0 to 9999 into the **Retry interval** box.

7. Specify an action to take if the job fails by choosing from the **On failure action** list.

8. If the job is a Transact-SQL script, you can choose from the following options:

    - Enter the name of an **Output file**. By default the file is overwritten each time the job step executes. If you do not want the output file overwritten, check **Append output to existing file**. This option is only available to members of the **sysadmin** fixed server role. Note that SQL Server Management Studio does not allow users to view arbitrary files on the file system, so you cannot use Management Studio to view job step logs that are written to the file system.

    - Check **Log to table** if you want to log the job step to a database table. By default the table contents are overwritten each time the job step executes. If you do not want the table contents overwritten, check **Append output to existing entry in table**. After the job step executes, you can view the contents of this table by clicking **View**.

    - Check **Include step output in history** if you want the output included in the step's history. Output will only be shown if there were no errors. Also, output may be truncated.

9. If you are a member of the **sysadmin** fixed server role and you want to run this job step as a different SQL login, select the SQL login from the **Run as user** list.

Using SQL Server Management Objects

To define Transact-SQL job step options

Use the **JobStep** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell.

Feedback

Was this page helpful? Yes No

Delete a Job Category

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to delete a Microsoft SQL Server Agent job category in SQL Server by using SQL Server Management Studio, Transact-SQL or SQL Server Management Objects.

Job categories help you organize your jobs for easy filtering and grouping. For example, you can organize all your database backup jobs in the Database Maintenance category.

Before You Begin

Limitations and Restrictions

When you delete a user-defined job category, SQL Server Agent prompts you to reassign the jobs that are assigned to it to another job category. You can only delete user-defined job categories.

Security

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To delete a job category

1. In **Object Explorer**, click the plus sign to expand the server where you want to delete a job category.

2. Click the plus sign to expand **SQL Server Agent**.

3. Right-click the **Jobs** folder and select **Manage Job Categories**.
4. In the **Manage Job Categories***server_name* dialog box, select the job category to delete.
5. Click **Delete**.
6. In the **Job Categories** dialog box, click **Yes**.
7. Close the **Manage Job Categories***server_name* dialog box.

Using Transact-SQL

To delete a job category

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- deletes the job category named AdminJobs.
USE msdb ;
GO
EXEC dbo.sp_delete_category
    @name = N'AdminJobs',
    @class = N'JOB' ;
GO
```

For more information, see sp_delete_category (Transact-SQL).

Using SQL Server Management Objects

To delete a job category

Call the **JobCategory** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell.

Feedback

Was this page helpful? Yes No

Delete a Job Step Log

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to delete a SQL Server Agent job step log.

- **Before you begin:**

  Limitations and Restrictions

  Security

- **To delete a SQL Server Agent job step log, using:**

  SQL Server Management Studio

  Transact-SQL

  SQL Server Management Objects

Before You Begin

Limitations and Restrictions

When job steps are deleted their output log is automatically deleted.

Security

Permissions

Unless you are a member of the **sysadmin** fixed server role, you can only modify jobs that you own.

Using SQL Server Management Studio

To delete a SQL Server Agent job step log

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**, expand **Jobs**, right-click the job you want to modify, and then click **Properties**.
3. In the **Job Properties** dialog box, delete the selected job step.

Using Transact-SQL

To delete a SQL Server Agent job step log

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- removes the job step log for step 2 in the job Weekly Sales Data
Backup
USE msdb ;
GO

EXEC dbo.sp_delete_jobsteplog
    @job_name = N'Weekly Sales Data Backup',
    @step_id = 2;
GO
```

For more information, see sp_delete_jobsteplog (Transact-SQL).

Using SQL Server Management Objects

Use the **DeleteJobStepLogs** methods of the **Job** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell. For more information, see SQL Server Management Objects (SMO).

```
-- Uses PowerShell to delete all job step log files that have ID values
larger than 5.
$srv = new-object Microsoft.SqlServer.Management.Smo.Server("(local)")
$jb = $srv.JobServer.Jobs["Test Job"]
$jb.DeleteJobStepLogs(5)
```

Feedback

Was this page helpful? Yes No

Delete a SQL Server Agent Proxy

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to delete a SQL Server Agent proxy account in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Limitations and Restrictions

- When you delete a SQL Server Agent proxy account, make sure the proxy does not reference any active job steps. To check for any job steps that reference the proxy, right-click the proxy, select **Properties**, and then, in the *proxy_name***Proxy Account Properties** dialog box, select the **References** page. If you delete a proxy, you are given the option to reassign all job steps that use that proxy in the **Delete Object** dialog box.
- SQL Server Agent proxies use credentials to store information about Windows user accounts. The user specified in the credential must have "Log on as a batch job" permission on the computer on which SQL Server is running.
- SQL Server Agent checks subsystem access for a proxy and gives access to the proxy each time the job step runs. If the proxy no longer has access to the subsystem, the job step fails. Otherwise, SQL Server Agent impersonates the user that is specified in the proxy and runs the job step.
- If the login for the user has access to the proxy, or the user belongs to any role with access to the proxy, the user can use the proxy in a job step.

Security

Permissions

Only members of the **sysadmin** fixed server role can create, modify, or delete proxy accounts.

Using SQL Server Management Studio

To delete a SQL Server Agent proxy account

1. In **Object Explorer**, click the plus sign to expand a server that contains the proxy account that you want to delete.
2. Click the plus sign to expand **SQL Server Agent**.
3. Click the plus sign to expand the **Proxies** folder.
4. Click the plus sign to expand the subsystem that contains the proxy account you want to delete (for example, **ActiveX Script)**.
5. Right-click the proxy account that you want to delete and select **Delete**.
6. In the **Delete Object** dialog box, confirm that the correct proxy account is selected. Check the **Reassign to** check box to reassign the job steps that reference this proxy account to another account.
7. Click **OK**.

Using Transact-SQL

To delete a SQL Server Agent proxy account

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- deletes the proxy "Catalog application proxy"
USE msdb ;
GO
EXEC dbo.sp_delete_proxy
```

```
    @proxy_name = N'Catalog application proxy' ;
GO
```

For more information, see sp_delete_proxy (Transact-SQL).

Feedback

Was this page helpful? Yes No

Delete an Alert

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This article describes how to delete Microsoft SQL Server Agent alerts using SQL Server Management Studio or Transact-SQL.

Before you begin

Limitations and restrictions

Removing an alert also removes any notifications associated with the alert.

Security

Permissions

By default, only members of the **sysadmin** fixed server role can delete alerts.

Using SQL Server Management Studio

To delete an alert

1. In **Object Explorer,** select the plus sign to expand the server that contains the SQL Server Agent alert that you want to delete.
2. Select the plus sign to expand **SQL Server Agent**.
3. Select the plus sign to expand the **Alerts** folder.
4. Right-click the alert you want to delete and select **Delete**.

5. In the **Delete Object** dialog box, confirm you chose the correct alert then select **OK**.

Using Transact-SQL

To delete an alert

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, select **New Query**.
3. Copy and paste the following example into the query window and select **Execute**.

```
-- deletes the SQL Server Agent alert called 'Test Alert.'
USE msdb ;
GO

EXEC dbo.sp_delete_alert
   @name = N'Test Alert' ;
GO
```

For more information, see sp_delete_alert (Transact-SQL).

Feedback

Was this page helpful? Yes No

Delete an Operator

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to remove an operator so they no longer receive Microsoft SQL Server Agent alert notifications in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Limitations and Restrictions

When an operator is removed, all the notifications associated with the operator are also removed.

Security

Permissions

Members of the **sysadmin** fixed server role can delete operators.

Using SQL Server Management Studio

To delete an operator

1. In **Object Explorer**, click the plus sign to expand the server that contains the operator you want to delete.
2. Click the plus sign to expand **SQL Server Agent**.
3. Click the plus sign to expand the **Operators** folder.

4. Right-click the operator that you want to delete and select **Delete**.

5. In the **Delete Object** dialog box, make sure that the correct operator is selected, and then click **OK**. If you want another operator to receive the alerts and jobs sent to the deleted operator, check **Reassign to** and select an operator in the list.

Using Transact-SQL

To delete an operator

1. In **Object Explorer**, connect to an instance of Database Engine.

2. On the Standard bar, click **New Query**.

3. Copy and paste the following example into the query window and click **Execute**.

```SQL
-- deletes operator 'Test Operator' and reassigns all alerts and jobs
--  sent to that operator to 'François Ajenstat'
USE msdb ;
GO

EXEC sp_delete_operator @name = 'Test Operator',
    @reassign_to_operator = 'François Ajenstat';
GO
```

For more information, see sp_delete_operator (Transact-SQL).

Feedback

Was this page helpful? Yes No

Delete Jobs

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

A job is a specified series of operations performed sequentially by SQL Server Agent. By default, jobs are not deleted when execution finishes. You can delete one or more Microsoft SQL Server Agent jobs regardless of success or failure of the job. You can also configure Microsoft SQL Server Agent to automatically delete jobs when they succeed, fail, or complete.

By default, members of the **sysadmin** fixed server role can execute the sp_delete_job (Transact-SQL) system stored procedure to delete a job. Other users must be granted one of the following SQL Server Agent fixed database roles in the **msdb** database:

- **SQLAgentUserRole**
- **SQLAgentReaderRole**
- **SQLAgentOperatorRole**

For details about the permissions of these roles, see SQL Server Agent Fixed Database Roles.

Members of the **sysadmin** fixed server role can execute **sp_delete_job** to delete any job. A user that is not a member of the **sysadmin** fixed server role can only delete jobs owned by that user.

Related Tasks

Expand table

Description	Topic
Describes how to delete one or more Microsoft SQL Server Agent jobs.	Delete One or More Jobs

Description	Topic
Describes how to configure Microsoft SQL Server Agent to automatically delete jobs when they succeed, fail, or complete.	Automatically Delete a Job

Feedback

Was this page helpful? Yes No

Delete One or More Jobs

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to delete Microsoft SQL Server Agent jobs in SQL Server by using SQL Server Management Studio, Transact-SQL, or SQL Server Management Objects.

Before You Begin

Security

Unless you are a member of the **sysadmin** fixed server role, you can only delete jobs that you own.

Using SQL Server Management Studio

To delete a job

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**, expand **Jobs**, right-click the job you want to delete, and then click **Delete**.
3. In the **Delete Object** dialog box, confirm that the job you intend to delete is selected.
4. Click **OK**.

To delete multiple jobs

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**.
3. Right-click **Job Activity Monitor**, and click **View Job Activity**.
4. In the Job Activity Monitor, select the jobs you want to delete, right-click your selection, and choose **Delete jobs**.

Using Transact-SQL

To delete a job

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
USE msdb ;
GO

EXEC sp_delete_job
    @job_name = N'NightlyBackups' ;
GO
```

For more information, see sp_delete_job (Transact-SQL).

Using SQL Server Management Objects

To delete multiple jobs

Use the **JobCollection** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell. For more information, see SQL Server Management Objects (SMO).

Feedback

Was this page helpful? Yes No

Delete Operator

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

Use this page to delete an operator.

Options

Object to be deleted
Shows the operator to be deleted.

Re-assign to
Reassigns notifications for the operator to be deleted.

Properties
Displays properties of the operator to reassign notifications to.

Related content

- Operators

Feedback

Was this page helpful? Yes No

Designate a Fail-Safe Operator

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

A fail-safe operator is a user who receives the alert if the designated operator cannot be reached. This topic describes how to set a fail-safe operator to receive Microsoft SQL Server Agent alert notifications in SQL Server by using SQL Server Management Studio.

Before You Begin

Limitations and Restrictions

- The Pager and **net send** options will be removed from SQL Server Agent in a future version of Microsoft SQL Server. Avoid using these features in new development work, and plan to modify applications that currently use these features.
- Note that SQL Server Agent must be configured to use Database Mail to send e-mail and pager notifications to operators. For more information, see Assign Alerts to an Operator.
- SQL Server Management Studio provides an easy, graphical way to manage jobs, and is the recommended way to create and manage the job infrastructure.

Security

Permissions

Only members of the **sysadmin** fixed server role can designate fail-safe operators.

Using SQL Server Management Studio

To designate a fail-safe operator

1. In **Object Explorer,** click the plus sign to expand the server that contains the SQL Server Agent operator that you want to designate as a fail-safe.
2. Right-click **SQL Server Agent** and select **Properties**.
3. In the **SQL Server Agent Properties -***server_name* dialog box, under **Select a page**, select **Alert System**.
4. Under **Fail-safe operator**, select **Enable fail-safe operator**.
5. In the **Operator** list, select the operator that you want to make a fail-safe.
6. Select either any or all of the following check boxes to specify how the operator will be notified: **E-mail**, **Pager**, or **Net send**.
7. When finished, click **OK**.

Feedback

Was this page helpful? Yes No

Designate an Events Forwarding Server

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance from SQL Server for details.

This topic describes how to designate a server to which Microsoft SQL Server forwards events in SQL Server by using SQL Server Management Studio . Note that event forwarding applies to events forwarded between servers, not to events forwarded between instances of SQL Server hosted on a single computer. Also note that in order to receive forwarded events, the alerts management server must be a default instance of SQL Server.

Before You Begin

Security

Permissions

Requires membership in the **sysadmin** fixed server role.

Using SQL Server Management Studio

To designate an events forwarding server

1. In **Object Explorer,** click the plus sign to expand the server from which you want to forward events to another server.
2. Right-click **SQL Server Agent** and select **Properties**.
3. In the **SQL Server Agent Properties** -*server_name* dialog box, under **Select a page**, select **Advanced**.

4. Under **SQL Server event forwarding**, select the **Forward events to a different server** check box.
5. In the **Server** list, select a server, and then, under **Events**, select one of the following:
    - Select **Unhandled events** to forward only the events that have not been handled by local alerts.
    - Select **All events** to forward all events regardless of whether they have been handled by local alerts.
6. In the **If event has severity at or above** list, click the severity level at which events are forwarded to the selected server.
7. When finished, click **OK**.

Feedback

Was this page helpful? Yes No

Disable or Enable a Job

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance from SQL Server for details.

This topic describes how to disable a SQL Server Agent job in SQL Server by using SQL Server Management Studio or Transact-SQL. When you disable a job, it is not deleted and can be enabled again when necessary.

Before You Begin

Security

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To disable or enable a job

1. In **Object Explorer**, connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**.
3. Expand **Jobs**, and then right-click the job that you want to disable or enable.
4. To disable a job, click **Disable**. To enable a job, click **Enable**.

Using Transact-SQL

To disable or enable a job

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- changes the name, description, and disables status of the job
NightlyBackups.
USE msdb ;
GO

EXEC dbo.sp_update_job
    @job_name = N'NightlyBackups',
    @new_name = N'NightlyBackups -- Disabled',
    @description = N'Nightly backups disabled during server
migration.',
    @enabled = 0 ;
GO
```

For more information, see sp_update_job (Transact-SQL).

Feedback

Was this page helpful? Yes No

Disable or Reactivate an Alert

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to disable or reactivate a Microsoft SQL Server Agent alert in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Security

Permissions

By default, members of the **sysadmin** fixed server role can edit information in an alert. Other users must be granted the **SQLAgentOperatorRole** fixed database role in the **msdb** database.

Using SQL Server Management Studio

To disable or reactivate an alert

1. In **Object Explorer**, click the plus sign to expand server that contains the alert you wish to disable or reactivate.
2. Click the plus sign to expand **SQL Server Agent**.
3. Click the plus sign to expand the **Alerts** folder.
4. Right-click the alert you wish to enable and select **Enable** To disable an alert, right-click the alert you want to disable and select **Disable**.

5. The **Disable Alert** or **Enable Alert** dialog box displays showing the status of the process. When finished, click **Close**.

Using Transact-SQL

To disable or reactivate an alert

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- changes the enabled setting of Test Alert to 0
USE msdb ;
GO

EXEC dbo.sp_update_alert
    @name = N'Test Alert',
    @enabled = 0 ;
GO
```

For more information, see sp_update_alert (Transact-SQL).

Feedback

Was this page helpful? Yes No

Edit an Alert

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to edit a Microsoft SQL Server Agent alert in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Security

Permissions

By default, members of the **sysadmin** fixed server role can edit information in an alert. Other users must be granted the **SQLAgentOperatorRole** fixed database role in the **msdb** database.

Using SQL Server Management Studio

To edit an alert

1. In **Object Explorer,** click the plus sign to expand the server containing the alert you want to edit.
2. Click the plus sign to expand **SQL Server Agent**.
3. Click the plus sign to expand the **Alerts** folder.
4. Right-click the alert you want to edit and select **Properties**.
5. Update the alert properties on the **General**, **Response**, and **Options** pages.

6. When finished, click **OK**.

Using Transact-SQL

To edit an alert

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- changes the enabled setting of Test Alert to 0
USE msdb ;
GO

EXEC dbo.sp_update_alert
    @name = N'Test Alert',
    @enabled = 0 ;
GO
```

For more information, see sp_update_alert (Transact-SQL).

Feedback

Was this page helpful? Yes No

Edit an Operator

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to edit an operators' availability for receiving notifications and their e-mail, pager, and net send addresses in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Limitations and Restrictions

- The Pager and **net send** options will be removed from SQL Server Agent in a future version of Microsoft SQL Server. Avoid using these features in new development work, and plan to modify applications that currently use these features.
- Note that SQL Server Agent must be configured to use Database Mail to send e-mail and pager notifications to operators. For more information, see Assign Alerts to an Operator.
- SQL Server Management Studio provides an easy, graphical way to manage jobs, and is the recommended way to create and manage the job infrastructure.

Security

Permissions

Only members of the **sysadmin** fixed server role can edit operators.

Using SQL Server Management Studio

To edit an operator

1. In **Object Explorer**, click the plus sign to expand the server that contains the operator you want to edit.
2. Click the plus sign to expand **SQL Server Agent**.
3. Click the plus sign to expand the **Operators** folder.
4. Right-click the operator that you want to edit and select **Properties**.

   For more information on the available options contained in the *operator_name***Properties** dialog box, see:

   - Operator Properties - New Operator (General Page)
   - Operator Properties - New Operator (Notifications Page)
   - Operator Properties (History Page)
5. When finished, click **OK**.

Using Transact-SQL

To edit an operator

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- updates the operator status to enabled, and sets the days
-- (from Monday through Friday, from 8 A.M. through 5 P.M.)
--  when the operator can be paged.
USE msdb ;
GO

EXEC dbo.sp_update_operator
    @name = N'François Ajenstat',
    @enabled = 1,
    @email_address = N'françoisa',
    @pager_address = N'5551290AW@pager.Adventure-Works.com',
    @weekday_pager_start_time = 080000,
    @weekday_pager_end_time = 170000,
```

```
    @pager_days = 64 ;
GO
```

For more information, see sp_update_operator (Transact-SQL).

Feedback

Was this page helpful? Yes No

Set a SQL Server Alias for the SQL Server Agent Service

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to set a Microsoft SQL Server alias for SQL Server Agent to use to connect to the Database Engine by using SQL Server Management Studio. By default, the SQL Server Agent service connects to an instance of SQL Server over named pipes by using dynamic server names that require no additional client configuration. You need to configure a server connection alias only if you are not using the default network transport or if you are connecting to an instance of SQL Server that listens on an alternate named pipe.

Before You Begin

Limitations and Restrictions

- SQL Server Agent will not work correctly unless you select an alias that refers to the local instance of SQL Server.
- Object Explorer only displays the SQL Server Agent node if you have permission to use it.

Security

Permissions

To perform its functions, SQL Server Agent must be configured to use the credentials of an account that is a member of the **sysadmin** fixed server role in SQL Server. The account must have the following Windows permissions:

- Log on as a service (SeServiceLogonRight)
- Replace a process-level token (SeAssignPrimaryTokenPrivilege)
- Bypass traverse checking (SeChangeNotifyPrivilege)
- Adjust memory quotas for a process (SeIncreaseQuotaPrivilege)

For more information about the Windows permissions required for the SQL Server Agent service account, see Select an Account for the SQL Server Agent Service and Setting Up Windows Service Accounts.

Using SQL Server Management Studio

To set a SQL Server Alias for the SQL Server Agent Service

1. In the **Object Explorer**, connect to an instance of the SQL Server Database Engine and then expand that instance.
2. Right-click **SQL Server Agent**, and then click **Properties**.
3. In the **SQL Server Agent Properties***server_name* dialog box, under **Select a page**, select **Connection**, and
4. In the **Alias local host server** box, type the alias of the server to which SQL Server Agent should connect.
5. Click **OK**.

Feedback

Was this page helpful? Yes No

Set CPU Idle Time and Duration

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic explains how to define the CPU idle condition for your server in SQL Server by using SQL Server Management Studio. The CPU idle definition influences how Microsoft SQL Server Agent responds to events. For example, suppose that you define the CPU idle condition as when the average CPU usage falls below 10 percent and remains at this level for 10 minutes. Then if you have defined jobs to execute whenever the server CPU reaches an idle condition, the job will start when the CPU usage falls below 10 percent and remains at that level for 10 minutes. If this is a job that significantly impacts the performance of your server, how you define the CPU idle condition is important.

Using SQL Server Management Studio

To set CPU idle time and duration

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Right-click **SQL Server Agent**, click **Properties**, and select the **Advanced** page.
3. Under **Idle CPU condition**, do the following:
    - Check **Define idle CPU condition.**
    - Specify a percentage for the **Average CPU usage falls below** (across all CPUs) box. This sets the usage level that the CPU must fall below before it is considered idle.
    - Specify a number of seconds for the **And remains below this level for** box. This sets the duration that the minimum CPU usage must remain at before it is considered idle.

Feedback

Was this page helpful? Yes No

Set Encryption Options on Target Servers

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

If you cannot use a certificate for Transport Layer Security (TLS), previously known as Secure Sockets Layer (SSL), encrypted communications between master servers and some or all of your target servers, but you want to encrypt the channel between them, configure the target server to use the level of security required.

To configure the appropriate level of security required for a specific master server/target server communication channel, set the SQL Server Agent registry subkey **\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server** *<instance_name>* **\SQLServerAgent\MsxEncryptChannelOptions(REG_DWORD)** on the target server to one of the following values. The value of *<instance_name>* is **MSSQL.***n*. For example, **MSSQL.1** or **MSSQL.3**.

Expand table

Value	Description
0	Disables encryption between this target server and the master server. Choose this option only when the channel between the target server and master server is secured by another means.
1	Enables encryption only between this target server and the master server, but no certificate validation is required.
2	Enables full TLS encryption and certificate validation between this target server and the master server. This setting is the default. Unless you have specific reason to choose a different value, we recommend not changing it.

If **1** or **2** is specified, you must have TLS enabled on both the master and target servers. If **2** is specified, you must also have a properly signed certificate present on the master server.

> **⊗ Caution**
>
> Incorrectly editing the registry can severely damage your system. Before making changes to the registry, we recommend that you back up any valued data on the computer.

See Also

How to: Enable Encrypted Connections to the Database Engine (SQL Server Configuration Manager)

Feedback

Was this page helpful? Yes No

Set Job Execution Shutdown

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to set the time that Microsoft SQL Server Agent will wait for executing jobs to finish before SQL Server Agent itself finishes in SQL Server by using SQL Server Management Studio.

Before You Begin

Security

Permissions

By default, members of the **sysadmin** fixed server role can set the time that Microsoft SQL Server Agent will wait for executing jobs to finish before SQL Server Agent itself finishes. Other users must be granted one of the following SQL Server Agent fixed database roles in the **msdb** database:

- **SQLAgentUserRole**
- **SQLAgentReaderRole**
- **SQLAgentOperatorRole**

Using SQL Server Management Studio

To set job execution shutdown

1. In **Object Explorer,** , click the plus sign to expand the server where you want to set a job execution shutdown interval.

2. Right-click **SQL Server Agent** and select **Properties**.
3. Under **Select a page**, select **Job System**.
4. Set the **Shutdown time-out interval** in seconds to increase or decrease the shutdown time-out interval. This determines how long SQL Server Agent will wait for executing jobs to finish before SQL Server Agent itself finishes.

Feedback

Was this page helpful? Yes No

Set job step success or failure flow

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

When creating Microsoft SQL Server Agent jobs, you can specify what action SQL Server should take if a failure occurs during job execution. Determine the action that SQL Server should take upon the success or failure of each job step. Then use the following procedure to configure the job step action flow logic by using SQL Server Agent.

Before you begin

Security

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To set job step success or failure flow

1. In **Object Explorer**, expand **SQL Server Agent**, and then expand **Jobs**.
2. Right-click the job you want to edit, and then select **Properties**.
3. Select the **Steps** page, select a step, and then select **Edit**.
4. In the **Job Step Properties** dialog box, select the **Advanced** page.
5. In the **On success action** list, select the action to perform if the job step completes successfully.
6. In the **Retry attempts** box, enter the number of times from 0 through 9999 that the job step should be repeated before it's considered to have failed. If you entered a value greater than 0 in the **Retry attempts** box, enter in the **Retry**

interval (minutes) box the number of minutes from 1 through 9999 that must pass before the job step is retried.

7. In the **On failure action** list, select the action to perform if the job step fails.

8. If the job is a Transact-SQL script, you can choose from the following options:

   - In the **Output file** box, enter the name of an output file to which the script output will be written. By default the file is overwritten each time the job step executes. If you don't want the output file overwritten, check **Append output to existing file**.

   - Check **Log to table** if you want to log the job step to a database table. By default the table contents are overwritten each time the job step executes. If you don't want the table contents overwritten, check **Append output to existing entry in table**. After the job step executes, you can view the contents of this table by clicking **View**.

   - Check **Include step output in history** if you want the output included in the step's history. Output will only be shown if there were no errors. Also, output may be truncated.

9. If the **Run as user** list is available, select the proxy account with the credentials that the job will use.

Using Transact-SQL

To set job step success or failure flow

1. In **Object Explorer**, connect to an instance of Database Engine.

2. On the Standard bar, select **New Query**.

3. Copy and paste the following example into the query window and select **Execute**.

```
USE msdb;
GO
EXEC sp_add_jobstep
    @job_name = N'Weekly Sales Data Backup',
    @step_name = N'Set database to read only',
    @subsystem = N'TSQL',
    @command = N'ALTER DATABASE SALES SET READ_ONLY',
```

```
    @on_success_action = 1;
GO
```

For more information, see sp_add_jobstep (Transact-SQL).

Using SQL Server Management Objects

To set job step success or failure flow

Use the **JobStep** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell. For more information, see SQL Server Management Objects (SMO).

See also

- Download SQL Server Management Studio (SSMS)

Feedback

Was this page helpful? Yes No

Set the Polling Interval for Target Servers

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to set the frequency that Microsoft SQL Server Agent refreshes information from the master to the target servers. A job is a specified series of actions that SQL Server Agent performs. A multiserver job is a job that a master server runs on one or more target servers.

- **Before you begin:** Security
- **To set the polling interval for target servers, using:** SQL Server Management Studio, Transact-SQL

Before You Begin

Each target server can run one instance of the same job at the same time. Each target server periodically polls the master server, downloads a copy of any new jobs assigned to the target server, and then disconnects. The target server runs the job locally and then reconnects to the master server to upload the job outcome status.

> **Note**
>
> If the master server is inaccessible when the target server tries to upload job status, the job status is spooled until the master server can be accessed.

Security

For detailed information, see Implement SQL Server Agent Security and Choose the Right SQL Server Agent Service Account for Multiserver Environments.

Using SQL Server Management Studio

To set the polling interval for target servers

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Right-click **SQL Server Agent**, point to **Multi Server Administration**, and then click **Manage Target Servers**.
3. On the **Target Server Status** tab, click **Post Instructions**.
4. In the **Instruction type** list, select **Set polling interval**.
5. In the **Polling interval** box, enter the number of seconds from 10 through 28,800 that must pass before the target server polls the master server.
6. Under **Recipients**, do one of the following:
   a. Click **All target servers** if all target servers share the same polling interval.
   b. Click **These target servers** if not all target servers share the same polling interval, and then select each target server that will use this polling interval.

Using Transact-SQL

To set the polling interval for target servers

1. In Object Explorer, connect to an instance of the Database Engine, and then expand that instance.
2. On the toolbar, click **New Query**.
3. In the query window, use the sp_post_msx_operation (Transact-SQL) system stored procedure to set the polling interval for target servers.

See Also

sysdownloadlist

Feedback

Was this page helpful? Yes No

Set the Service Startup Account for SQL Server Agent (SQL Server Configuration Manager)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

The SQL Server Agent service startup account defines the Windows account that SQL Server Agent runs as, as well as its network permissions. This article describes how to set the SQL Server Agent service account with SQL Server Configuration Manager in SQL Server by using SQL Server Management Studio.

Limitations

By default, the SQL Server Agent service account is mapped to the default SQL Server Agent service SID (`NT SERVICE\SQLSERVERAGENT`), which is a member of the **sysadmin** fixed server role. The account must also be a member of the `msdb` database role **TargetServersRole** on the master server if multiserver job processing is used. The Master Server Wizard automatically adds the service account to this role as part of the enlistment process.

Object Explorer only displays the SQL Server Agent node if you have permission to use it.

Permissions

To perform its functions, SQL Server Agent must be configured to use the credentials of an account that is a member of the **sysadmin** fixed server role in SQL Server. The account must have the following Windows permissions:

- Log on as a service (`SeServiceLogonRight`)
- Replace a process-level token (`SeAssignPrimaryTokenPrivilege`)

- Bypass traverse checking (`SeChangeNotifyPrivilege`)
- Adjust memory quotas for a process (`SeIncreaseQuotaPrivilege`)

For more information about the Windows permissions required for the SQL Server Agent service account, see Select an Account for the SQL Server Agent Service and Setting Up Windows Service Accounts.

Use SQL Server Management Studio

Set the Service Startup Account for SQL Server Agent

1. In **Registered Servers**, select the plus sign to expand **Database Engine**.
2. Select the plus sign to expand the **Local Server Groups** folder.
3. Right-click the server instance where you want to set up the Service Startup Account, and select **SQL Server Configuration Manager...**.
4. In the **User Account Control** dialog box, select **Yes**.
5. In SQL Server Configuration Manager, in the console pane, select **SQL Server Services**.
6. In the details pane, right-click **SQL Server Agent***(server_name)*, where *server_name* is the name of the SQL Server Agent instance for which you want to change the service startup account, and select **Properties**.
7. In the **SQL Server Agent***(server_name)* **Properties** dialog box, in the **Log On** tab, select one of the following options under **Log on as**:

   - **Built-in account**: select this option if your jobs require resources from the local server only. For information about how to choose a Windows built-in account type, see Selecting an Account for SQL Server Agent Service.

     > ⓘ **Important**
     >
     > The SQL Server Agent service does not support the **Local Service** account in SQL Server Management Studio.

   - **This account**: select this option if your jobs require resources across the network, including application resources; if you want to forward events to

other Windows application logs; or if you want to notify operators through e-mail or pagers.

If you select this option:

a. In the **Account Name** box, enter the account that will be used to run SQL Server Agent. Alternately, select **Browse** to open the **Select User or Group** dialog box and select the account to use.

b. In the **Password** box, enter the password for the account. Re-enter the password in the **Confirm password** box.

8. Select **OK**.

9. In SQL Server Configuration Manager, select the **Close** button.

Feedback

Was this page helpful? Yes No

Set the SQL Server Connection for the SQL Server Agent Service

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to set the connection between SQL Server Agent and the Database Engine in SQL Server by using SQL Server Management Studio. The SQL Server Agent service can connect to a local instance of SQL Server by using Windows Authentication.

Before You Begin

Limitations and Restrictions

- Object Explorer only displays the SQL Server Agent node if you have permission to use it.
- Beginning with SQL Server 2005 (9.x), SQL Server Agent does not support SQL Server Authentication. This option is available only when you administer an earlier version of SQL Server.

Security

Permissions

To perform its functions, SQL Server Agent must be configured to use the credentials of an account that is a member of the **sysadmin** fixed server role in SQL Server. The account must have the following Windows permissions:

- Log on as a service (SeServiceLogonRight)
- Replace a process-level token (SeAssignPrimaryTokenPrivilege)

- Bypass traverse checking (SeChangeNotifyPrivilege)
- Adjust memory quotas for a process (SeIncreaseQuotaPrivilege)

For more information about the Windows permissions required for the SQL Server Agent service account, see Select an Account for the SQL Server Agent Service and Setting Up Windows Service Accounts.

Using SQL Server Management Studio

To set the SQL Server connection

1. In **Object Explorer**, click the plus sign to expand the server that you want to set up with a connection to its SQL Server Agent Service.
2. Right-click **SQL Server Agent** and select **Properties**.
3. In the **SQL Server Agent Properties** dialog box, under **Select a page**, click **Connection**.
4. Under **SQL Server connection**, select **Use Windows Authentication** to enable SQL Server Agent to connect to an instance of the SQL Server Database Engine with Microsoft Windows Authentication. Connections to SQL Server 2005 (9.x) and later databases require Windows Authentication.

Feedback

Was this page helpful? Yes No

Set Up the Job History Log

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to set up the Microsoft SQL Server Agent job history log.

- **Before you begin:** Security
- **To setup the job history log, using:** SQL Server Management Studio

Before You Begin

Security

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To set up the job history log

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Right-click **SQL Server Agent**, and then click **Properties**.
3. In the **SQL Server Agent Properties** dialog box, select the **History** page.
4. Choose from the following options:
    a. Check **Limit size of job history log**, and then type the maximum number of rows for the job history log, and the maximum number of rows per job.
    b. Check **Automatically remove agent history**, and specify a time period, such that history older than this period will be purged from the log.

See Also

Implement Jobs
Monitor Job Activity
Create Jobs

Feedback

Was this page helpful? Yes No

Start a Job

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This article describes how to start running a Microsoft SQL Server Agent job in SQL Server by using SQL Server Management Studio, Transact-SQL or SQL Server Management Objects.

A job is a specified series of actions that SQL Server Agent performs. SQL Server Agent jobs can run on one local server or on multiple remote servers.

Before You Begin

Security

For detailed information, see Implement SQL Server Agent Security.

Use SQL Server Management Studio

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent,** and expand **Jobs**. Depending on how you want the job to start, do one of the following:
   - If you're working on a single server, or working on a target server, or running a local server job on a master server, right-click the job you want to start, and then select **Start Job**.
   - If you want to start multiple jobs, right-click **Job Activity Monitor**, and then select **View Job Activity**. In the Job Activity Monitor you can select multiple jobs, right-click your selection, and select **Start Jobs**.

- If you're working on a master server and want all targeted servers to run the job simultaneously, right-click the job you want to start, select **Start Job**, and then select **Start on all targeted servers**.
- If you're working on a master server and want to specify target servers for the job, right-click the job you want to start, select **Start Job**, and then select **Start on specific target servers**. In the **Post Download Instructions** dialog box, select the **These target servers** check box, and then select each target server on which this job should run.

Use Transact-SQL

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, select **New Query**.
3. Copy and paste the following example into the query window and select **Execute**.

```SQL
-- starts a job named Weekly Sales Data Backup.
USE msdb ;
GO

EXEC dbo.sp_start_job N'Weekly Sales Data Backup' ;
GO
```

For more information, see sp_start_job (Transact-SQL).

Use programming languages

Call the **Start** method of the **Job** class by using a programming language that you choose, such as Visual Basic, Visual C#, or SQL PowerShell.

SQL PowerShell

Here's a PowerShell script that can be used in SQL Server Agent with parameters. This script demonstrates how to start a SQL Server Agent job using parameters passed into the script.

```PowerShell
# Parameters
param(
```

```
    [string]$ServerInstance,
    [string]$JobName
)

# Load the SMO assembly
Add-Type -AssemblyName "Microsoft.SqlServer.SMO"

# Create a server object
$server = New-Object Microsoft.SqlServer.Management.Smo.Server
$ServerInstance

# Get the job you want to start
$job = $server.JobServer.Jobs[$JobName]

# Start the job
if ($job) {
    $job.Start()
    Write-Output "The job '$JobName' on server '$ServerInstance' has been
started successfully."
} else {
    Write-Output "The job '$JobName' was not found on server
'$ServerInstance'."
}
```

How to use the script in SQL Server Agent.

1. Open SQL Server Management Studio (SSMS).
2. Connect to the appropriate SQL Server instance.
3. Expand the SQL Server Agent node.
4. Right-click on Jobs and select New Job.
5. In the New Job dialog box, enter the job name and other required details.
6. Go to the Steps page and select New to create a new job step.
7. In the New Job Step dialog box:
    a. Set the Type to PowerShell.
    b. In the Command field, enter the PowerShell script along with the parameters, for example:

    PowerShell

    ```
    .\YourScript.ps1 -ServerInstance "YourServerInstance" -JobName
    "YourJobName"
    ```

8. Set any other job properties as required (Schedules, Alerts, Notifications, etc.).

9. Select OK to save the job.

Explanation of the script

- Parameters: The script accepts two parameters, $ServerInstance and $JobName, which are the SQL Server instance and the job name respectively.
- Load SMO: The Add-Type cmdlet is used to load the SQL Server Management Objects (SMO) assembly.
- Server Object: A new server object is created using the $ServerInstance parameter.
- Get Job: The script retrieves the specified job using the $JobName parameter.
- Start Job: If the job is found, it's started using the Start method. The script outputs a success message. If the job isn't found, an error message is displayed.

For more information, see SQL Server Management Objects (SMO).

Related content

- Create a Job
- Install SQL PowerShell

Feedback

Was this page helpful? Yes No

Start, stop, or pause the SQL Server Agent service

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This article describes how to start, stop, or restart the SQL Server Agent Service in SQL Server by using SQL Server Management Studio.

You can configure the SQL Server Agent service to start automatically when the operating system starts, or you can start it manually when you need to complete jobs. You can stop or pause the SQL Server Agent service to suspend jobs, operator notifications, and alerts.

Limitations

Microsoft SQL Server Agent must be running as a service in order to automate administrative tasks. For more information, see Configure SQL Server Agent.

Object Explorer only displays the SQL Server Agent node if you have permission to use it.

Permissions

To perform its functions, SQL Server Agent must be configured to use the credentials of an account that is a member of the **sysadmin** fixed server role in SQL Server. The account must have the following Windows permissions:

- Log on as a service (`SeServiceLogonRight`)
- Replace a process-level token (`SeAssignPrimaryTokenPrivilege`)
- Bypass traverse checking (`SeChangeNotifyPrivilege`)
- Adjust memory quotas for a process (`SeIncreaseQuotaPrivilege`)

For more information about the Windows permissions required for the SQL Server Agent service account, see Select an account for the SQL Server Agent service and Configure Windows service accounts and permissions.

Use SQL Server Management Studio

1. In **Object Explorer**, select the plus sign to expand the server where you want to manage SQL Server Agent Service.
2. Right-click **SQL Server Agent**, and then select either **Start**, **Stop**, or **Restart**.
3. In the **User Account Control** dialog box, select **Yes**.
4. When prompted if you want to perform the action, select **Yes**.

Related content

- Start, stop, pause, resume, and restart SQL Server services
- Auto Restart SQL Server Agent

Feedback

Was this page helpful? Yes No

Stop a Job

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to stop a Microsoft SQL Server Agent job. A job is a specified series of actions that SQL Server Agent performs.

- **Before you begin:** ,

  Limitations and Restrictions

  Security

- **To stop a job, using:**

  SQL Server Management Studio

  Transact-SQL

  SQL Server Management Objects

Before You Begin

Limitations and Restrictions

- If a job is currently executing a step of type **CmdExec** or **PowerShell**, the process that is being run (for example, MyProgram.exe) is forced to end prematurely. This can cause unpredictable behavior, such as files that are being used by the process being held open.
- For a multiserver job, a STOP instruction for the job is posted to all target servers of the job.

Security

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To stop a job

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**, expand **Jobs**, right-click the job you want to stop, and then click **Stop Job**.
3. If you want to stop multiple jobs, right-click **Job Activity Monitor**, and then click **View Job Activity**. In the Job Activity Monitor, select the jobs you want to stop, right-click your selection, and then click **Stop Jobs**.

Using Transact-SQL

To stop a job

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- stops a job named Weekly Sales Data Backup
USE msdb ;
GO

EXEC dbo.sp_stop_job
    N'Weekly Sales Data Backup' ;
GO
```

For more information, see sp_stop_job (Transact-SQL).

Using SQL Server Management Objects

To stop a job

Call the **Stop** method of the **Job** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell. For more information, see SQL Server Management Objects (SMO).

Feedback

Was this page helpful? Yes No

View a Job

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to view Microsoft SQL Server Agent jobs in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Security

You can only view jobs that you own, unless you are a member of the **sysadmin** fixed server role. Members of this role can view all jobs. For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To view a job

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**, and then expand **Jobs**.
3. Right-click a job, and then click **Properties**.

Using Transact-SQL

To view a job

1. In **Object Explorer**, connect to an instance of Database Engine.

2. On the Standard bar, click **New Query**.

3. Copy and paste the following example into the query window and click **Execute**.

```
-- lists all aspects of the information for the job NightlyBackups.
USE msdb ;
GO

EXEC dbo.sp_help_job
    @job_name = N'NightlyBackups',
    @job_aspect = N'ALL' ;
GO
```

Using SQL Server Management Objects

To view a job

Use the **Job** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell. For more information, see SQL Server Management Objects (SMO).

Feedback

Was this page helpful? Yes No

View Information About an Alert

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to view information about Microsoft SQL Server Agent alerts in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Security

Permissions

By default, members of the **sysadmin** fixed server role can view information about an alert. Other users must be granted the **SQLAgentOperatorRole** fixed database role in the **msdb** database.

Using SQL Server Management Studio

To view information about an alert

1. In **Object Explorer,** click the plus sign to expand the server where you want to view information about an alert.
2. Click the plus sign to expand **SQL Server Agent**.
3. Click the plus sign to expand the **Alerts** folder.
4. Right-click the alert that has the information you want to view and select **Properties**.

For more information on the available options contained in the *alert_name***alert properties** dialog box, see:

- Alert Properties - New Alert (General Page)
- Alert Properties - New Alert (Response Page)
- Alert Properties - New Alert (Options Page)
- Alert Properties (History Page)

5. When finished, click **OK**.

Using Transact-SQL

To view information about an alert

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- reports information about the Demo: Sev. 25 Errors alert
-- This example assumes that the 'Demo: Sev. 25 Errors' alert exists.
USE msdb ;
GO

EXEC sp_help_alert @alert_name = 'Demo: Sev. 25 Errors'
GO
```

For more information, see sp_help_alert (Transact-SQL).

Feedback

Was this page helpful? Yes No

View Information About an Operator

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to view information about a Microsoft SQL Server Agent operator in SQL Server by using SQL Server Management Studio or Transact-SQL.

Before You Begin

Security

Permissions

By default, members of the **sysadmin** fixed server role can execute this stored procedure. Other users must be granted one of the following SQL Server Agent fixed database roles in the **msdb** database:

- **SQLAgentUserRole**
- **SQLAgentReaderRole**
- **SQLAgentOperatorRole**

For details about the permissions of these roles, see SQL Server Agent Fixed Database Roles.

Using SQL Server Management Studio

To view information about an operator

1. In **Object Explorer**, click the plus sign to expand the server that contains the operator you want to view.

2. Click the plus sign to expand **SQL Server Agent**.
3. Click the plus sign to expand the **Operators** folder.
4. Right-click the operator that you want to view and select **Properties**.

   For more information on the available options contained in the *operator_name***Properties** dialog box, see:

   - Operator Properties - New Operator (General Page)
   - Operator Properties - New Operator (Notifications Page)
   - Operator Properties (History Page)
5. When finished, click **OK**.

Using Transact-SQL

To view information about an operator

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- reports information about operator François Ajenstat
-- This example assumes that the operator exists.
USE msdb ;
GO

EXEC dbo.sp_help_operator
    @operator_name = N'François Ajenstat' ;
GO
```

For more information, see sp_help_operator (Transact-SQL).

Feedback

Was this page helpful? Yes No

View Job Activity

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to view the runtime state of SQL Server Agent jobs in SQL Server by using SQL Server Management Studio or Transact-SQL.

When the Microsoft SQL Server Agent service starts, a new session is created and the **sysjobactivity** table in the **msdb** database is populated with all the existing defined jobs. This table records current job activity and status. You can use the Job Activity Monitor in SQL Server Agent to view the current state of jobs. If the SQL Server Agent service unexpectedly terminates, you can refer to the **sysjobactivity** table to see which jobs were being executed when the service terminated.

Before You Begin

Security

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To view job activity

1. In **Object Explorer**, connect to an instance of the SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**.
3. Right-click **Job Activity Monitor** and click **View Job Activity**.

4. In the **Job Activity Monitor**, you can view details about each job that is defined for this server.

5. Right-click a job to start it, stop it, enable or disable it, refresh its status as displayed in the Job Activity Monitor, delete it, or view its history or properties. To start, stop, enable or disable, or refresh multiple jobs, select multiple rows in the Job Activity Monitor, and right-click your selection.

6. To update the Job Activity Monitor, click **Refresh**. To view fewer rows, click **Filter** and enter filter parameters.

Using Transact-SQL

To view job activity

1. In **Object Explorer**, connect to an instance of Database Engine.

2. On the Standard bar, click **New Query**.

3. Copy and paste the following example into the query window and click **Execute**.

```
-- lists activity for all jobs that the current user has permission to
view.
USE msdb ;
GO

EXEC dbo.sp_help_jobactivity ;
GO
```

For more information, see sp_help_jobactivity (Transact-SQL).

Feedback

Was this page helpful? Yes No

View Job Step Information

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to view job step details in the Job Step Properties dialog. It also includes information about viewing job step output.

- **Before you begin:**

  Limitations and Restrictions

  Security

- **To view job step information, using:**

  SQL Server Management Studio

Before You Begin

Limitations and Restrictions

If the job step has been configured to write its output to a table or file and the job has run at least once, you can view its output on the **Advanced** page of the **Job Step Properties** dialog. When a job or job step is deleted, the output log is automatically deleted.

Security

Permissions

You can view only those jobs that you own, unless you are a member of the **sysadmin** fixed server role. Members of this role can view all jobs and job step details.

Using SQL Server Management Studio

To view job step information

1. In **Object Explorer,** connect to an instance of the Microsoft SQL Server Database Engine, and then expand that instance.
2. Expand **SQL Server Agent**, expand **Jobs**, right-click the job that contains the job step to be viewed, and click **Properties**.
3. In the **Job Properties** dialog, click the **Steps** page.
4. Click the job step to be viewed, and click **Edit**.
5. On the **General** page of the **Job Step Properties** dialog, you can view the type of job step and what it does.
6. Click the **Advanced** page to view the actions SQL Server Agent takes if the job step succeeds or fails, how many times the job step should be attempted, where the job step output is written, and the user the job step runs as.

To view job step output

1. In the **Job Step Properties** dialog, click the **Advanced** page.
2. Depending on the version of SQL Server you are connected to, you can view either the job step output file or table as follows:
   - When you are connected to SQL Server or later, you can click **View** only when **Log to table** is checked. In this case, the job step output is written to the **sysjobstepslogs** table in the **msdb** database.
   - The **View** button is disabled when job step output is written to a file. To view a job step output file, use Notepad.

Feedback

Was this page helpful? Yes No

View or Modify Jobs

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

You can view any job you have created. After you have run a job, you can also view its history. Viewing a job's history allows you to see when the job ran, the status of the job as a whole, and the status of each job step in the job. You can see whether the job ever failed in the past, when the job last completed successfully, and what output the job created each time the job ran. Members of the **sysadmin** fixed server role can view or modify any job, regardless of the owner.

> **Note**
>
> A job must have been executed at least one time for there to be a job history. You can limit the total size of the job history log and the size per job.

Finally, you can modify a job to meet new requirements. You can modify:

- Response options
- Schedules
- Job steps
- Ownership
- Job category
- Target servers (multiserver jobs only)

To make sure that changes to multiserver jobs take effect, you must post the changes to the download list so that target servers can download the updated job. To ensure that target servers have the most current job definitions, post an INSERT instruction after you update the multiserver job as follows:

```
EXECUTE sp_post_msx_operation 'INSERT', 'JOB', '<job id>'
```

For more information, see sp_purge_jobhistory (Transact-SQL).

Members of the **sysadmin** fixed server role can view the definition or history of any job, and can modify any job.

Related Tasks

Expand table

Description	Topic
Describes how to view Microsoft SQL Server Agent jobs.	View a Job
Describes how to view the Microsoft SQL Server Agent job history log.	View the Job History
Describes how to delete the contents of the Microsoft SQL Server Agent job history log.	Clear the Job History Log
Describes how to set size limits for Microsoft SQL Server Agent job history logs.	Resize the Job History Log
Describes how to change the properties of Microsoft SQL Server Agent jobs.	Modify a Job

See Also

sysjobhistory

Feedback

Was this page helpful? Yes No

View the Job History

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to view the Microsoft SQL Server Agent job history log in SQL Server by using SQL Server Management Studio, Transact-SQL, or SQL Server Management Objects.

- **Before you begin:**

  Security

- **To view the job history log, using:**

  SQL Server Management Studio

  Transact-SQL

  SQL Server Management Objects

Before You Begin

Security

For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To view the job history log

1. In **Object Explorer,** connect to an instance of the SQL Server Database Engine, and then expand that instance.

2. Expand **SQL Server Agent**, and then expand **Jobs**.
3. Right-click a job, and then click **View History**.
4. In the Log File Viewer, view the job history.
5. To update the job history, click **Refresh**. To view fewer rows, click the **Filter** button and enter filter parameters.

Using Transact-SQL

To view the job history log

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- lists all job information for the NightlyBackups job.
USE msdb ;
GO

EXEC dbo.sp_help_jobhistory
    @job_name = N'NightlyBackups' ;
GO
```

For more information, see sp_help_jobhistory (Transact-SQL).

Using SQL Server Management Objects

To view the job history log

Call the **EnumHistory** method of the **Job** class by using a programming language that you choose, such as Visual Basic, Visual C#, or PowerShell. For more information, see SQL Server Management Objects (SMO).

Feedback

Was this page helpful?

Yes No

SQL Server Agent Error Log

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

SQL Server Agent creates an error log that records warnings and errors by default. The following warnings and errors are displayed in the log:

- Warning messages that provide information about potential problems, such as "Job *<job_name>* was deleted while it was running."
- Error messages that usually require intervention by a system administrator, such as "Unable to start mail session." Error messages can be sent to a specific user or computer by **net send**.

SQL Server maintains up to nine SQL Server Agent error logs. Each archived log has an extension that indicates the relative age of the log. For example, an extension of .1 indicates the newest archived error log and an extension of .9 indicates the oldest archived error log.

By default, execution trace messages are not written to the SQL Server Agent error log, because they can fill it. When the error log is full, your ability to select and analyze more difficult errors is reduced. Because the log adds to the server's processing load, it is important to consider carefully what value you obtain by capturing execution trace messages to the error log. Generally, it is best to capture all messages only when you are debugging a specific problem.

When SQL Server Agent is stopped, you can modify the location of the SQL Server Agent error log. When the error log is empty, the log cannot be opened. You can cycle the SQL Server Agent log at any time without stopping SQL Server Agent using dbo.sp_cycle_agent_errorlog.

To view the SQL Server Agent error log

- View SQL Server Agent Error Log (SQL Server Management Studio)

To rename a SQL Server Agent error log

- Rename a SQL Server Agent Error Log (SQL Server Management Studio)

To send SQL Server Agent error messages

- Send SQL Server Agent Error Messages

To write execution trace messages to the SQL Server Agent error log

- Write Execution Trace Messages to the SQL Server Agent Error Log (SQL Server Management Studio)

Feedback

Was this page helpful? Yes No

View SQL Server Agent Error Log

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to view the SQL Server Agent error log in SQL Server by using SQL Server Management Studio.

Log File Viewer displays log information from many different components. When Log File Viewer is open, use the **Select logs** pane to select the logs you want to display. Each log displays columns appropriate to that kind of log. The logs that are available depend on how Log File Viewer is opened.

Before You Begin

Limitations and Restrictions

Object Explorer only displays the SQL Server Agent node if you have permission to use it.

Security

Permissions

To perform its functions, SQL Server Agent must be configured to use the credentials of an account that is a member of the **sysadmin** fixed server role in SQL Server. The account must have the following Windows permissions:

- Log on as a service (SeServiceLogonRight)
- Replace a process-level token (SeAssignPrimaryTokenPrivilege)
- Bypass traverse checking (SeChangeNotifyPrivilege)

- Adjust memory quotas for a process (SeIncreaseQuotaPrivilege)

For more information about the Windows permissions required for the SQL Server Agent service account, see Selecting an Account for SQL Server Agent Service and Setting Up Windows Service Accounts.

Using SQL Server Management Studio

To view the SQL Server Agent error log

1. In **Object Explorer**, click the plus sign to expand the server that contains the SQL Server Agent error log that you want to view.

2. Click the plus sign to expand **SQL Server Agent**.

3. Click the plus sign to expand the **Error Logs** folder.

4. Right-click the error log you want to view and select **View Agent Log**.

   The following options are available in the **Log File Viewer** -*server_name* dialog box:

   Load Log
   Open a dialog box where you can specify a log file to load.

   Export
   Open a dialog box that lets you export the information that is shown in the **Log file summary** grid to a text file.

   Refresh
   Refresh the view of the selected logs. The **Refresh** button rereads the selected logs from the target server while applying any filter settings.

   Filter
   Open a dialog box that lets you specify settings that are used to filter the log file, such as **Connection**, **Date**, or other **General** filter criteria.

   Search
   Search the log file for specific text. Searching with wildcard characters is not supported.

   Stop
   Stops loading the log file entries. For example, you can use this option if a remote or offline log file takes a long time to load, and you only want to view the most recent entries.

Log file summary
This information panel displays a summary of the log file filtering. If the file is not filtered, you will see the following text, **No filter applied**. If a filter is applied to the log, you will see the following text, **Filter log entries where:** <filter criteria>.

Selected row details
Select a row to display additional details about the selected event row at the bottom of the page. The columns can be reordered by dragging them to new locations in the grid. The columns can be resized by dragging the column separator bars in the grid header to the left or right. Double-click the column separator bars in the grid header to automatically size the column to the content width.

Instance
The name of the instance on which the event occurred. This is displayed as *computer name**instance name*.

Date
Displays the date of the event.

Source
Displays the source feature from which the event is created, such as the name of the service (MSSQLSERVER, for example). This does not appear for all log types.

Message
Displays any messages associated with the event.

Log Type
Displays the type of log to which the event belongs. All selected logs appear in the log file summary window.

Log Source
Displays a description of the source log in which the event is captured.

5. When finished, click **Close**.

Feedback

Was this page helpful? Yes No

Rename a SQL Server Agent Error Log

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to rename the file where Microsoft SQL Server Agent errors are written in SQL Server by using SQL Server Management Studio.

Before You Begin

Limitations and Restrictions

- Object Explorer only displays the SQL Server Agent node if you have permission to use it.
- SQL Server Agent will not write to the new log file until the SQL Server Agent service is restarted.

Security

Permissions

To perform its functions, SQL Server Agent must be configured to use the credentials of an account that is a member of the **sysadmin** fixed server role in SQL Server. The account must have the following Windows permissions:

- Log on as a service (SeServiceLogonRight)
- Replace a process-level token (SeAssignPrimaryTokenPrivilege)
- Bypass traverse checking (SeChangeNotifyPrivilege)
- Adjust memory quotas for a process (SeIncreaseQuotaPrivilege)

For more information about the Windows permissions required for the SQL Server Agent service account, see Select an Account for the SQL Server Agent Service and Setting Up Windows Service Accounts.

Using SQL Server Management Studio

To rename a SQL Server Agent error log

1. In **Object Explorer**, click the plus sign to expand the server that contains the SQL Server Agent error log that you want to rename.
2. Click the plus sign to expand **SQL Server Agent**.
3. Right-click the **Error Logs** folder and select **Configure**.
4. In the **Configure SQL Server Agent Error Logs** dialog box, in the **Error log file** box, enter the new file path and file name for the error log. Alternately, click the ellipsis **(...)** to open the **Specify agent error log location** dialog box.
5. When finished, click **OK**.

Feedback

Was this page helpful? Yes No

Write Execution Trace Messages to the SQL Server Agent Error Log

Article • 02/13/2025

Applies to: SQL Server Azure SQL Managed Instance

> **Important**
>
> On Azure SQL Managed Instance, most, but not all SQL Server Agent features are currently supported. See Azure SQL Managed Instance T-SQL differences from SQL Server for details.

This topic describes how to configure Microsoft SQL Server Agent to include execution trace messages in its error log in SQL Server by using SQL Server Management Studio.

Before You Begin

Limitations and Restrictions

- Object Explorer only displays the SQL Server Agent node if you have permission to use it.
- Because this option can cause the error log to become large, only include execution trace messages in SQL Server Agent error logs when investigating a specific SQL Server Agent problem.

Security

Permissions

To perform its functions, SQL Server Agent must be configured to use the credentials of an account that is a member of the **sysadmin** fixed server role in SQL Server. The account must have the following Windows permissions:

- Log on as a service (SeServiceLogonRight)
- Replace a process-level token (SeAssignPrimaryTokenPrivilege)
- Bypass traverse checking (SeChangeNotifyPrivilege)

- Adjust memory quotas for a process (SeIncreaseQuotaPrivilege)

For more information about the Windows permissions required for the SQL Server Agent service account, see Select an Account for the SQL Server Agent Service and Setting Up Windows Service Accounts.

To write execution trace messages to the SQL Server Agent error log

1. In **Object Explorer**, click the plus sign to expand the server that contains the SQL Server Agent error log to which you want to write execution trace messages.
2. Right-click **SQL Server Agent** and select **Properties**.
3. In the **SQL Server Agent Properties** -*server_name* dialog box, under **Error log** on the **General** page, select the **Include execution trace messages** check box.
4. Click **OK**.

Feedback

Was this page helpful? Yes No

Create a New Registered Server (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server

This article describes how to save the connection information for servers that you access frequently, by registering the server in the Registered Servers component of SQL Server Management Studio in SQL Server. A server can be registered before connecting, or when connecting from Object Explorer. There's a special menu option to register the server instances on the local computer.

There are two kinds of registered servers:

- Local server groups

  Use local server groups to easily connect to servers that you frequently manage. Both local and nonlocal servers are registered into local server groups. Local server groups are unique to each user. For information about how to share registered server information, see Export Registered Server Information (SQL Server Management Studio) and Import Registered Server Information (SQL Server Management Studio).

  > **Note**
  >
  > We recommend that you use Windows Authentication whenever possible.

- Central Management Servers

  Central Management Servers store server registrations in the Central Management Server instead of on the file system. Central Management Servers and subordinate registered servers can be registered only by using Windows Authentication. After a Central Management Server has been registered, its associated registered servers will be automatically displayed. For more information about Central Management Servers, see Administer Multiple Servers Using Central Management Servers. Versions of SQL Server that are earlier than SQL Server 2008 (10.0.x) can't be designated as a Central Management Server.

Using SQL Server Management Studio

To create a new registered server

1. If Registered Servers isn't visible in SQL Server Management Studio, on the **View** menu, select **Registered Servers**.

   Server type
   When a server is registered from Registered Servers, the **Server type** box is read-only, and matches the type of server displayed in the Registered Servers pane. To register a different type of server, select **Database Engine**, **Analysis Server**, **Reporting Services**, or **Integration Services** on the **Registered Servers** toolbar before starting to register a new server.

   Server name
   Select the server instance to register in the format: <*servername*>[\<*instancename*>].

   Authentication
   Two authentication modes are available when connecting to an instance of SQL Server.

   Windows Authentication
   Windows Authentication mode allows a user to connect through a Microsoft Windows user account.

   SQL Server Authentication
   When a user connects with a specified login name and password from a nontrusted connection, SQL Server performs the authentication itself by checking whether a SQL Server login account has been set up and whether the specified password matches the one previously recorded. If SQL Server doesn't have a login account set, authentication fails, and the user receives an error message.

   > ⓘ **Important**
   >
   > When possible, use Windows authentication. For more information, see Choose an Authentication Mode.

   User name
   Shows the current user name you're connecting with. This read-only option is only available if you have selected to connect using Windows Authentication. To change **User names**, sign into the computer as a different user.

   Login
   Enter the login to connect with. This option is available only if you have selected to

connect using SQL Server Authentication.

Password
Enter the password for the login. This option can be edited only if you have selected to connect by using SQL Server Authentication.

Remember password
Select to have SQL Server encrypt and store the password you have entered. This option is displayed only if you have selected to connect using SQL Server Authentication.

> ⓘ **Note**
>
> If you have stored the password and want to stop storing it, clear this check box, and then click **Save**.

Registered server name
The name you want to appear in Registered Servers. This name doesn't have to match the **Server name** box.

Registered server description
Enter an optional description of the server.

Test
Select to test the connection to the server selected in **Server name**.

Save
Select to save the registered server settings.

Multiserver Queries

The Query Editor window in SQL Server Management Studio can connect to and query multiple instances of SQL Server at the same time. The results that are returned by the query can be merged into a single results pane, or they can be returned in separate results panes. As an option, Query Editor can include columns that provide the name of the server that produced each row, and also the login that was used to connect to the server that provided each row. For more information about how to execute multiserver queries, see Execute Statements Against Multiple Servers Simultaneously (SQL Server Management Studio).

To execute queries against all the servers in a local server group, right-click the server group, select **Connect**, and then select **New Query**. When queries are executed in the

new Query Editor window, they'll execute against all servers in the group, using the stored connection information including the user authentication context. Servers registered by using SQL Server Authentication but not saving the password will fail to connect.

To execute queries against all the servers that are registered with a Central Management Server, expand the Central Management Server, right-click the server group, select **Connect**, and then select **New Query**. When queries are executed in the new Query Editor window, they'll execute against all of the servers in the server group, using the stored connection information and using the Windows Authentication context of the user.

See also

- Hide System Objects in Object Explorer
- Export Registered Server Information (SQL Server Management Studio)
- Import Registered Server Information (SQL Server Management Studio)
- Administer multiple servers using Central Management Servers

---

Feedback

Was this page helpful? Yes No

Register a Connected Server (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes how to register a connected server in SQL Server by using SQL Server Management Studio (SSMS). By registering the server, you can save the connection information for servers that you access frequently. A server can be registered before connecting, or at the time of connection from Object Explorer. You can view your registered servers in SSMS by navigating to **View\Registered Servers** from the menu.

In This Topic

- **To register a server, using:**

  SQL Server Management Studio

Using SQL Server Management Studio

To register a connected server

In Object Explorer, right-click a server to which you already are connected, and then click **Register**.

Server name
This field defaults to the server name you connected to. Optionally, you can type a server name or choose one from the drop-down list.

Authentication
Two authentication modes are available when connecting to an instance of SQL Server.

- **Windows Authentication**
  Windows Authentication mode allows a user to connect through a Microsoft Windows user account.

- **SQL Server Authentication**
  When a user connects with a specified login name and password from a nontrusted connection, SQL Server performs the authentication itself by checking whether a SQL Server login account has been set up and whether the specified

password matches the one previously recorded. If SQL Server does not have a login account set, authentication fails, and the user receives an error message.

> **ⓘ Important**
>
> When possible, use Windows authentication. For more information, see Choose an Authentication Mode.

- **User name**
  Shows the current user name you are connecting with. This read-only option is only available if you have selected to connect using Windows Authentication. To change **User names**, log in to the computer as a different user.
- **Login**
  Enter the login to connect with. This option is available only if you have selected to connect using SQL Server Authentication.
- **Password**
  Enter the password for the login. This option can be edited only if you have selected to connect by using SQL Server Authentication.
- **Remember password**
  Select to have SQL Server encrypt and store the password you have entered. This option is displayed only if you have selected to connect using SQL Server Authentication.

  > **ⓘ Note**
  >
  > If you have stored the password and want to stop storing it, clear this check box, and then click **Save**.

Registered server name
The name you want to appear in Registered Servers. This name does not have to match the **Server name** box.

Registered server description
Enter an optional description of the server.

Test
Click to test the connection to the server selected in **Server name**.

Save
Click to save the registered server settings.

See Also

Create a New Registered Server (SQL Server Management Studio)

Feedback

Was this page helpful? Yes No

Create a Central Management Server and Server Group

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes how to designate an instance of SQL Server as a central management server in SQL Server by using SQL Server Management Studio. Central management servers store a list of instances of SQL Server that is organized into one or more central management server groups. Actions that are taken by using a central management server group act on all servers in the server group. This includes connecting to servers by using Object Explorer and executing Transact-SQL statements and Policy-Based Management policies on multiple servers at the same time.

> **Note**
>
> Versions of SQL Server that are earlier than SQL Server 2008 (10.0.x) cannot be designated as a central management server.

In This Topic

- **Before you begin:**

  Security

- **To create a Central Management Server and Server Group, using:**

  SQL Server Management Studio

Before You Begin

Security

Permissions

Two database roles in the msdb database grant access to central management servers. Only members of the ServerGroupAdministratorRole role can manage the central

management server. Membership in the ServerGroupReaderRole role is required to connect to a central management server.

Because the connections that are maintained by a central management server execute in the context of the user, by using Windows Authentication, the effective permissions on the registered servers might vary. For example, the user might be a member of the sysadmin fixed server role on the instance of SQL Server A, but have limited permissions on the instance of SQL Server B.

Using SQL Server Management Studio

The following procedures describe how to perform the following steps.

1. Create a central management server.
2. Add one or more server groups to the central management server and add one or more registered servers to the server groups.

Create a central management server

1. In SQL Server Management Studio, on the **View** menu, click **Registered Servers**.
2. In Registered Servers, expand **Database Engine**, right-click **Central Management Servers**, and then click **Register Central Management Server**.
3. In the **New Server Registration** dialog box, select the instance of SQL Server that you want to become the central management server from the drop-down list of servers. You must use Windows Authentication for the central management server.
4. In **Registered Server**, enter a server name and optional description.
5. From the **Connection Properties** tab, review or modify the network and connection properties. For more information, see Connect to Server (Connection Properties Page) Database Engine
6. Click **Test**, to test the connection.
7. Click **Save**. The instance of SQL Server will appear under the **Central Management Servers** folder.

Create a new server group and add servers to the group

1. From **Registered Servers**, expand **Central Management Servers**. Right-click the instance of SQL Server added in the procedure above and select **New Server Group**.

2. In **New Server Group Properties**, enter a group name and optional description.

3. From **Registered Servers**, right-click the server group and click **New Server Registration**.

4. From New Server Registration, select an instance of SQL Server. For more information, see Create a New Registered Server (SQL Server Management Studio). Add more servers as appropriate.

To execute queries against several configuration targets at the same time

- After you create a central management server, one or more server groups, and one or more registered servers, you can execute queries against a whole group at the same time. For more information about how to execute Transact-SQL statements on the servers in a server group at the same time, see Execute Statements Against Multiple Servers Simultaneously (SQL Server Management Studio).

See Also

Administer Multiple Servers Using Central Management Servers

Feedback

Was this page helpful? Yes No

Execute statements against multiple servers simultaneously

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This article describes how to query multiple servers at the same time in SQL Server, by creating a local server group, or a Central Management Server and one or more server groups, and one or more registered servers within the groups, and then querying the complete group.

The results returned by the query can be combined into a single results pane, or can be returned in separate results panes. The result set can include additional columns for the server name and the login used by the query on each server. Central Management Servers and subordinate servers can be registered by using only Windows Authentication. Servers in local server groups can be registered by using Windows Authentication or SQL Server authentication.

Before you execute the following procedures, create a Central Management Server and server group. For more information, see Create a Central Management Server and Server Group (SQL Server Management Studio).

Permissions

Because the connections maintained by a Central Management Server execute in the context of the user, by using Windows Authentication, the effective permissions on the registered servers might vary. For example, the user might be a member of the **sysadmin** fixed server role on the instance of SQL Server A, but have limited permissions on the instance of SQL Server B.

Execute statements against multiple configuration targets simultaneously

1. In SQL Server Management Studio, on the **View** menu, select **Registered Servers**.
2. Expand a Central Management Server, right-click a server group, point to **Connect**, and then select **New Query**.
3. In Query Editor, type and execute a Transact-SQL statement, such as the following:

```
SQL

USE master
GO
SELECT * FROM sys.databases;
GO
```

By default, the results pane will combine the query results from all the servers in the server group.

Change the multiserver results options

1. In Management Studio, on the **Tools** menu, select **Options**.
2. Expand **Query Results**, expand **SQL Server**, and then select **Multiserver Results**.
3. On the **Multiserver Results** page, specify the option settings that you want, and then select **OK**.

See also

- Administer Multiple Servers Using Central Management Servers

---

Feedback

Was this page helpful? Yes No

View Registered Servers in SQL Server Management Studio

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes how to view registered servers in SQL Server.

To view Registered Servers in SQL Server Management Studio

1. In SQL Server Management Studio, on the **View** menu, click **Registered Servers**.

   By default, Registered Servers appears above Object Explorer in the main window.

Feedback

Was this page helpful? Yes No

Remove a Registered Server (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This article describes how to remove a server from Registered Servers in SQL Server by using SQL Server Management Studio.

To remove a registered server

1. Select **View** > **Registered Servers**.
2. In the Registered Servers Object Explorer pane, right-click a server, and then select **Delete**.
3. In the **Delete Confirmation** dialog box, select **Yes**.

Feedback

Was this page helpful? Yes No

Change a Server's Registration (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes how to change a server's registration in SQL Server by using SQL Server Management Studio. You can change the connection information for a registered server at any time.

To change a server's registration

1. In Registered Servers, right-click a server, and then click **Properties**.
2. In the **Edit Server Registration Properties** dialog box, modify the server, login information, or connection properties, and then click **Save**.

See Also

Register Servers

Feedback

Was this page helpful? Yes No

Connect to a Registered Server (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes how to connect to a registered server in SQL Server by using SQL Server Management Studio. If the server is not running or cannot be found on the network, Registered Servers displays an error.

To connect to a registered server

1. If the Registered Servers pane is not visible, select **View** > **Registered Servers**.
2. In Registered Servers, right-click a server, point to **Connect**, and then select **Object Explorer**.

Feedback

Was this page helpful? Yes No

Disconnect from a Registered Server (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes how to disconnect Object Explorer from a server at any time in SQL Server. Disconnecting Object Explorer does not disconnect other SQL Server Management Studio components such as SQL Editor.

To disconnect from a registered server

- In Object Explorer, right-click a server and click **Disconnect** or, on the Object Explorer toolbar, click the **Disconnect** icon.

Feedback

Was this page helpful? Yes No

Move a Registered Server or Registered Server Group

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This article describes how to organize the servers in Registered Servers by moving a registered server or server groups in SQL Server. Server groups can contain registered servers, or other server groups. Both servers and server groups can be moved from one server group to another.

To move a registered server or server group

1. In Registered Servers, right-click a server or server group, and then select **Tasks** > **Move To**.
2. In the **Move Server Registration** dialog box, expand the list of server groups, select the node where you want the server or server group to appear, and then select **OK**.

Related content

- Register Servers
- Create or Edit a Server Group (SQL Server Management Studio)

Feedback

Was this page helpful? Yes No

Change the Name of Registered Server or Registered Server Group

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes how to change the name of a registered server or server group in SQL Server by using SQL Server Management Studio. The name can be changed at any time. Changing the name of a server in Registered Servers only changes how the name is displayed. To connect to a different server, you must edit the connection properties of the registered server.

Using SQL Server Management Studio

From the menu navigate to **View\Registered Servers** to open the **Registered Servers** pane.

To change the name of a server

1. In **Registered Servers**, expand **Database Engine** and then **Local Server Groups**.
2. Right-click a server and select **Properties** to open the **Edit Server Registration Properties** dialog window.
3. In the **Registered server name** text box, type the new name for the server registration, and then click **Save**.

To change the name of a server group

1. In **Registered Servers**, expand **Database Engine** and then **Local Server Groups**.
2. Right-click a server group and select **Properties** to open the **Edit Server Group Properties** dialog window.
3. In the **Group name** text box, type the new name for the server group, and then click **Save**.

See Also

Change a Server's Registration (SQL Server Management Studio)

Feedback

Was this page helpful? Yes No

Create or Edit a Server Group (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes how to organize the servers in Registered Servers in SQL Server by creating server groups, and placing the servers in the server groups. You can create server groups in Registered Servers at any time, or you can create server groups when you register servers.

To create a server group in Registered Servers

1. In Registered Servers, click the server type on the Registered Servers toolbar. If Registered Servers is not visible, click **Registered Servers** on the **View** menu.
2. Right-click a server or a server group, point to **New**, and then click **Server Group**.
3. In the **New Server Group** dialog box, in the **Group name** list box, type a unique name for the server group. The server group name must be unique for the current location in the Registered Servers tree.
4. In the **Group description** list box, optionally type a friendly name that describes the server group, for example, "Finance Servers for Latin America."
5. In the **Select a location for the new server group** box, click a location for the group, and then click **Save**.

> **Note**
>
> You can also create a new server group while you are registering a server by clicking **New Group**, and then completing the **New Group** dialog box.

See Also

Register Servers

Feedback

Was this page helpful? Yes No

Remove a Server Group (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes how to remove a server group in Registered Servers in SQL Server by using SQL Server Management Studio. You can delete a server group at any time. If the server group is not empty, any servers or server groups contained in the deleted server group will also be deleted. Before deleting a server group, move any servers or server groups that you want to retain to a new server group.

To remove a server group

1. In Registered Servers, right-click a server group, and then click **Delete**.
2. In the **Delete Confirmation** dialog box, click **Yes**.

See Also

Move a Registered Server or Registered Server Group (SQL Server Management Studio)

Feedback

Was this page helpful? Yes No

Export Registered Server Information (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes how to save and export registered server information in SQL Serverand distribute it to other employees or servers. You can use this export feature to present a consistent user interface on multiple computers.

Exporting and then importing Registered Server files lets you easily configure several computers with the same servers in Registered Servers. This is useful when managing a large number of servers from computers in several locations, or when you want to configure a less experienced user with basic connection settings.

> **Note**
>
> Server registrations that use SQL Server Authentication store passwords on a per-user basis. After exporting and importing the server registrations, users must enter the password for each server the first time they connect, storing the passwords in their lists of registered servers. This is not necessary for servers registered using Windows Authentication.

To export registered server information

1. In Registered Servers, right-click a server group, and then select **Tasks** > **Export...**.

   > **Note**
   >
   > You can export an individual server, all of the registered server tree, or a subset of the registered server tree.

2. In the **Export Registered Servers** dialog box, make the following selections.

   Server group
   Specify the server group which will be exported. Export all registered servers, registered servers in a particular server group, or a single registered server to the export file. The export functionality is recursive; for example, if server group A

contains server group B, and server group B contains server groups C and D, exporting server group A exports all entries in A, B, C, and D.

The server group displays only the server groups of the current registered server tree.

Export file
Type the path and name of the export file in the text box or use the Browse button (...) to select a location on the client computer, and name for the export file. Use the `.regsrvr` extension for the file. If you select an existing file, the registered server information is appended to the file. If you want your registered server information to be available to other users or another computer, you can save the file on the network. Other users can access the file and import part or all of the registered server information. If you select an existing file as your export file, the contents of the file are overwritten with the server registration information.

Do not include user names and passwords in the export file
Exclude user names when exporting the file.

> **Important**
>
> Although the export files are encrypted, if user names and SQL Server Authentication passwords are included in the file, access to the file should be carefully controlled. User names and passwords are therefore excluded from the export file by default.

See Also

Import Registered Server Information (SQL Server Management Studio)
Create a New Registered Server (SQL Server Management Studio)

Feedback

Was this page helpful? Yes No

Import registered server information (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server, Azure SQL Database, Azure SQL Managed Instance, Azure Synapse Analytics, Analytics Platform System (PDW)

This article describes how to import saved registered server information in SQL Server. Exporting and then importing registered server files lets you easily configure several computers with the same servers in Registered Servers. This is useful when managing a large number of servers from computers in several locations, or when you want to configure basic connection settings for a less-experienced user.

> **Note**
>
> You can't import registered server information from earlier versions of SQL Server than the version you are currently using.

Use SQL Server Management Studio

1. In Registered Servers, select the server type on the Registered Servers toolbar. The server type must be the same as the registered server export file type. For example, if you exported SQL Server registered server information, you must select **SQL Server** on the Registered Servers toolbar.

   If Registered Servers isn't visible in SQL Server Management Studio, on the **View** menu, select **Registered Servers**.

2. Right-click a server group, and select **Tasks** > **Import...**.

3. In the **Import Registered Servers** dialog box, select the registered servers file to import, and then select **OK**.

   a. **Import file**

      Type the path and name of the import file in the text box, or select the Browse button (...) to locate the import file on the client computer. If you select an existing file, the registered server information is appended to the file. The import file can only be a previously exported registered server file. Registered server files have a `.regsrvr` extension.

b. **Select the server group to import to**

Select the root node or a particular server group to which the registered server entries in the file will be imported. You can import all registered servers, registered servers in a particular server group, or a single registered server to the export file. The import functionality is recursive. For example, if server group `A` contains server group `B`, and server group `B` contains server groups `C` and `D`, importing server group `A` exports all entries in `A`, `B`, `C`, and `D`.

The server group displays only the server groups of the current registered server tree.

If you select to import an existing server group or server, a message confirms that you want to overwrite the existing server or server group.

Server registrations that use SQL Server Authentication store passwords on a per-user basis. After you import the server registrations, you must enter the password for each server the first time you connect, storing the passwords in your lists of registered servers. This isn't necessary for servers registered through Windows Authentication.

Related content

- Change a Server's Registration (SQL Server Management Studio)
- Export Registered Server Information (SQL Server Management Studio)
- Create a New Registered Server (SQL Server Management Studio)

Feedback

Was this page helpful? Yes No

Database Engine Scripting

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The SQL Server Database Engine supports the Microsoft PowerShell scripting environment to manage instances of the Database Engine and the objects in the instances. You can also build and run Database Engine queries that contain Transact-SQL and XQuery in environments very similar to scripting environments.

SQL Server PowerShell

SQL Server includes two SQL Server PowerShell snap-ins that implement:

- A SQL Server PowerShell provider that exposes the SQL Server management object model hierarchies as PowerShell paths that are similar to file system paths. You can use the SQL Server management object model classes to manage the objects represented at each node of the path.
- A set of SQL Server cmdlets that implement SQL Server commands. One of the cmdlets is **Invoke-Sqlcmd**. This is used to run Database Engine Query scripts to be run with the **sqlcmd** utility.

SQL Server provides these features for running PowerShell:

- The **sqlps** PowerShell module that can be imported to a PowerShell session, the module then loads the SQL Server snap-ins. You can interactively run ad hoc PowerShell commands. You can run script files using a command such as .\MyFolder\MyScript.ps1.
- PowerShell script files can be used as input to SQL Server Agent PowerShell job steps that run the scripts either at scheduled intervals or in response to system events.
- The **sqlps** utility that starts PowerShell and imports the SQL Server module. You can then perform all actions supported by the module. You can start the **sqlps** utility either in a command prompt or by right-clicking on the nodes in the SQL Server Management Studio Object Explorer tree and selecting **Start PowerShell**.

Database Engine Queries

Database Engine query scripts contain three types of elements:

- Transact-SQL language statements.
- XQuery language statements
- Commands and variables from the **sqlcmd** utility.

SQL Server provides three environments for building and running Database Engine queries:

- You can interactively run and debug Database Engine queries in the Database Engine Query Editor in SQL Server Management Studio. You can code and debug several statements in one session, then save all of the statements in a single script file.
- The **sqlcmd** command prompt utility lets you interactively run Database Engine queries, and also run existing Database Engine query script files.

Database Engine query script files are typically coded interactively in SQL Server Management Studio by using the Database Engine Query Editor. The file can later be opened in one of these environments:

- Use the SQL Server Management Studio **File**/**Open** menu to open the file in a new Database Engine Query Editor window.
- Use the **-i***input_file* parameter to run the file with the **sqlcmd** utility.
- Use the **-QueryFromFile** parameter to run the file with the **Invoke-Sqlcmd** cmdlet in SQL Server PowerShell scripts.
- Use SQL Server Agent Transact-SQL job steps to run the scripts either at scheduled intervals or in response to system events.

In addition, you can use the SQL Server Generate Script Wizard to generate Transact-SQL scripts. You can right-click objects in the SQL Server Management Studio Object Explorer, then select the **Generate Script** menu item. **Generate Script** launches the wizard, which guides you through the process of creating a script.

Database Engine Scripting Tasks

Expand table

Task Description	Topic
Describes how to use the code and text editors in Management Studio to interactively develop, debug, and run Transact-SQL scripts	Query and Text Editors (SQL Server Management Studio)
Describes how to use the **sqlcmd** utility to run Transact-SQL scripts from the command prompt, including the ability to interactively develop scripts.	sqlcmd How-to Topics
Describes how to integrate the SQL Server components into a Windows PowerShell environment and then build PowerShell scripts for managing SQL Server instances and objects.	SQL Server PowerShell
Describes how to use the **Generate and Publish Scripts** wizard to create Transact-SQL scripts that recreate one or more of the objects from a database.	Generate Scripts (SQL Server Management Studio)

See Also

sqlcmd Utility
Tutorial: Writing Transact-SQL Statements

Feedback

Was this page helpful? Yes No

Query and Text Editors (SQL Server Management Studio)

Article04/01/2012

You can use one of the SQL Server Management Studio editors to interactively edit and test a Transact-SQL, MDX, DMX, or XML/A script, or to edit an XML or plain text file. Each editor is supported by a language-specific service that colors keywords, and checks for syntax and usage errors. The Database Engine Query Editor includes a Transact-SQL debugger that you can use to help fix problems in Transact-SQL code.

SQL Server Management Studio Editors

The four editors in Management Studio share a common architecture. The text editor implements the base level of functionality, and can be used as a basic editor for text files. The other three editors, or query editors, extend this base of functionality by including a language service that defines the syntax of one of the languages supported in SQL Server. The query editors also implement varying levels of support for editor features such as IntelliSense and debugging. The query editors include the Database Engine Query Editor for use in building scripts containing Transact-SQL and XQuery statements, the MDX editor for the MDX language, the DMX editor for the DMX language, and the XML/A editor for the XML for Analysis language.

Common Components

All of the editors in Management Studio share these components:

- **Code Pane**
  The area where you enter your queries or text. In the query editors, it contains the statement builder features available for your language. The text editing environment supports find and replace, bulk commenting, and custom fonts and colors.

  You can set options that affect the behavior of text in the code pane as it relates to indenting, tabbing, dragging and dropping of text, and so forth. Query windows can be configured to operate as either tabs in the document window, or in separate documents.

- **Selection Margin**
  A column of white space between the margin indicator bar and the code text where you can click to select lines of text. You can hide or display the selection margin.

- **Horizontal and Vertical Scroll Bars**
  Allows you to scroll the code pane horizontally and vertically so that you can view the

code that extends beyond the viewable edges of the code pane.

- **Line Numbering**
  Displays line numbers to the left of the text or code in the Editor. You can navigate to specific line numbers.

- **Word Wrap**
  Displays long lines of text or code as multiple lines, enabling you to see all the text on the line. Word wrap does not affect the way text appears when it is executed or printed. Word wrap is turned on from the **Tools**, **Options** dialog box, on either the Text Editor, All Languages, General page, or on a specific editor page.

Code Editor Components

The code editors contain these features in addition to the ones shared with the text and XML editors:

- **Results**
  This window is used to view the results of a query. The window can display the results in grid or in text, or the results can be directed to a file. Result grids can be displayed as separate tabbed windows.

- **IntelliSense**
  In the Editors, on the **Edit** menu, point to **IntelliSense**, to view the Microsoft IntelliSense options.

- **Color Coding**
  Displays different colors for each type of syntax element, which improves the readability of complex statements.

- **Code Outlining**
  Displays code groups with outlining lines to the left of the code. Code groups can be collapsed and expanded to make it easier to review your code.

- **Template**
  Templates are files that include the basic structure of the statements needed to create objects in a database. They can be used to speed the authoring of scripts.

- **Messages**
  Displays errors, warnings, and informational messages that are returned by the server when a script is run. The list of messages does not change until the script is run again.

- **Status Bar**
  Displays system information that is associated with the Query Editor window, such as

which instance the Query Editor is connected to.

Database Engine Query Editor Components

These components are only available in the Database Engine Query Editor:

- **Debugger**
  Enables you to pause the execution of code on specific statements. You can then view data and system information to help you find errors in the code.
- **Error List**
  Displays syntax and semantic errors found by IntelliSense. The list of errors changes dynamically as you edit Transact-SQL scripts.
- **Graphical Showplan**
  Displays the logical steps built into the execution plan of a Transact-SQL statement.
- **Client Statistics**
  Displays information about the query execution grouped into categories. When **Include Client Statistics** is selected from the **Query** menu, a **Client Statistics** window is displayed upon query execution. Statistics from successive query executions are listed along with the average values. Select **Reset Client Statistics** from the **Query** menu to reset the average.
- **Code Snippets**
  Templates you can use as a starting point when adding statements in the Database Engine Query Editor. You can insert the pre-defined snippets supplied with SQL Server, or add your own snippets.
- **SQLCMD Mode**
  Runs Transact-SQL scripts that include the set of commands supported by the sqlcmd utility. For more information, see sqlcmd How-to Topics.

Editor Tasks

Expand table

Task Description	Topic
Describes how to view and use the basic features in the Database Engine Query Editor.	Database Engine Query Editor (SQL Server Management Studio)

Task Description	Topic
Describes how to view and use the basic features in the MDX Query Editor.	MDX Query Editor (Analysis Services - Multidimensional Data)
Describes how to view and use the basic features in the DMX Query Editor.	DMX Query Editor (Analysis Services - Data Mining)
Describes how to view and use the basic features in the XML/A Editor.	XML Editor (SQL Server Management Studio)
Describes how to configure options for the various editors, such as line numbering and IntelliSense options.	Configure Editors (SQL Server Management Studio)
Describes the various ways you can open the editors in Management Studio.	Open an Editor (SQL Server Management Studio)
Describes how to manage the view mode, such as word wrap, splitting a window, or tabs.	Manage the Editor and View Mode
Describes how to set formatting options, such as hidden text or indentation.	Manage Code Formatting
Describes how to navigate through the text in an editor window by using features such as incremental search or go to.	Navigate Code and Text
Describes how to set color coding options for various classes of syntax, which makes it easier to read complex statements.	Color Coding in Query Editors
Describes how to use code outlining to hide parts of complex scripts that you are not currently working on.	Code Outlining
Describes how to drag text from one location in a script and drop it in a new location.	Drag and Drop Text
Describes how to do global search and replace, such as when changing column names.	Search and Replace
Describes how to set bookmarks in order to more easily find important pieces of code.	Manage Bookmarks
Describes how to print scripts or the results in a window or grid.	Print Code and Results
Describes how to use the sqlcmd features in the Database Engine Query Editor.	Edit SQLCMD Scripts with Query Editor
Describes how to use IntelliSense features such as auto-completing object names as you type them, or ensuring breakpoints are placed in valid locations.	IntelliSense (SQL Server Management Studio)

Task Description	Topic
Describes how to use code snippets in the Database Engine Query Editor. Snippets are templates for commonly used statements or blocks, and can be customized or extended to include site-specific snippets.	Transact-SQL Code Snippets
Describes how to use the Transact-SQL debugger to step through code and view debugging information such as the values in variables and parameters.	Transact-SQL Debugger
Describes how to set custom colors for different instances of the Database Engine, and have those colors set as the background of the status bar in Database Engine Query Editor windows.	Status Bar (Database Engine Query Editor)

See Also

Reference

SQL Server Management Studio Keyboard Shortcuts

XML Editor (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Provides a set of visual tools for working with XML Schemas, ADO.NET datasets, and XML documents. The XML Designer supports the XML Schema Definition (XSD) language defined by the World Wide Web Consortium (WC3). The designer does not support DTDs (document type definitions) or other XML schema languages, such as XDR (XML-Data Reduced).

To display the designer, add a dataset, XML Schema, or XML file to your project or open any of the file types listed in the table below.

> **Caution**
>
> There is no **Undo** command when working in Schema view. Plan your work carefully and save your files often.

The designer provides the following three views (or modes) to work on XML files, XML Schemas, and datasets:

Expand table

View	Description	File types supported
Schema	For visually creating and modifying XML Schemas and ADO.NET datasets.	.xsd
Data	For visually modifying XML data files in a structured data grid.	.xml
XML	For editing XML; the source editor provides color-coding and IntelliSense, including Complete Word and List Members.	.xml .xsd .xslt .wsdl.web.resx.tdl.wsf.hta.disco.vsdisco.config
ShowPlan	Displays xml query plans created using the SET SHOWPLAN_XML ON option.	.showplan

Schema View

Schema view provides a visual representation of the elements, attributes, types, and so on, that make up XML Schemas and ADO.NET datasets.

In Schema view you can construct schemas and datasets by dropping elements on the design surface from either the XML Schema tab of the Toolbox or from Server Explorer. Additionally, you can add elements to the designer by right-clicking the design surface and selecting Add from the shortcut menu.

In Schema view you can:

- Construct and modify existing XML Schemas and ADO.NET datasets
- Create and edit relationships between tables
- Create and edit keys
- Generate ADO.NET datasets from XML Schemas

> ⓘ **Note**
>
> The layout of elements in Schema view is stored in the .xsx file, which can be seen by clicking **Show All Files** in the Solution Explorer toolbar, and then expanding the .xsd file. If there is no .xsx file present, it means the .xsd file has never been opened in the XML Designer.

Customizing Schema View

The following features modify the visual layout of elements in Schema view:

- Zooming
- Expanding or collapsing of nested elements
- Automatically arranging layout of elements
- Resetting default state of collapsed elements

To expand hidden nested elements

- Click the plus icon on the bottom of the element.

To collapse nested elements

- Click the minus icon on the bottom-most element that you want to appear on the designer.

Data View

Data view provides a data grid that can be used to modify .xml files. Only the content (but not the tags and structure) in an XML file can be edited in Data view.

There are two separate areas in Data view: **Data Tables** and **Data**. The **Data Tables** area is a list of relations defined in the XML file, in the order of its nesting (from the outermost to the innermost). The **Data** area is a data grid that displays data based on the selection in the Data Tables area.

> ⓘ **Note**
>
> Newly created XML files contain no data and therefore cannot be displayed in Data view. There are also some instances of XML documents where data view cannot be invoked at all. Although the XML would be considered well formed, if it is not structured data trying to switch to Data view will generate the following message: "Although this document is well formed, it contains structure that Data View cannot display."

In Data view you can:

- Manually populate data tables
- Edit existing data tables
- Generate an XML Schema from an XML document

XML View

XML view provides an editor for editing raw XML and provides IntelliSense and color coding. Statement completion is available when working on .xsd files and .xml files that have an associated schema. Type < to initiate a tag and you will be presented with a list of elements that are valid at that location. After you type the element name and press the SPACEBAR, you will be presented with a list of attributes that the element supports.

> ⓘ **Note**

Microsoft IntelliSense options are not available on the toolbar. When in the XML Editor, to access the options, on the **Edit** menu, click **IntelliSense**.

SHOWPLAN view

Query plans can be saved in XML format when created using SET SHOWPLAN_XML ON option. Double-click a file with the .showplan extension to open the query plan.

See Also

Save an Execution Plan in XML Format

Feedback

Was this page helpful? Yes No

Configure Editors (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can customize the operation of the SQL Server Management Studio editors by configuring the options for each editor.

Setting Editor Options

Most of the editor options are set by using the **Tools** menu and selecting **Options...** to display an **Options** dialog. In the **Options** dialog, open the **Text Editor** node in the left pane to set code and text editing options. The nodes under Text Editor apply to specific editors:

1. **All Languages** - options set using this node apply to all of the Management Studio editors. You can override these settings by using the other nodes to set different options for a specific editor.
2. **Plain Text** - options set using this node apply to the MDX, DMX, and text editors.
3. **Transact-SQL** - options set using this node apply to the Database Engine Query Editor.
4. **XML** - options set using this node apply to the XML for Analysis editor.

Open the **Query Execution** or **Query Results** nodes to customize the execution of queries and how the results are displayed.

Editor Configuration Tasks

Expand table

Task Description	Topic
Describes how to specify that an editor be opened by double-clicking on a file with a specified extension in Windows Explorer.	Associate File Extensions to a Code Editor
Describes how to customize fonts to make code and text more	Change Font Color, Size, and

Task Description	Topic
readable.	Style
Describes how to view properties.	Use the Properties Window in Management Studio
Location of the F1 help pages for the editor options dialogs.	Query Options Pages F1 Help

Feedback

Was this page helpful? Yes No

Associate file extensions to a code editor

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Associating file extensions to a specific code editor allows you to open a file with the appropriate SQL Server Management Studio code editor when you double-click a file in Windows Explorer. The extensions commonly used by Management Studio, such as .sql and .mdx, are associated during installation. New file extensions must also be associated to Management Studio in the file system. You can use this feature to open files created with other editors, or to open files you have renamed, such as backups of .sql files that were renamed .bak.

There are two steps in the process. First associate the extension with Management Studio, and then associate the extension with a specific code editor.

To associate a new file extension with SQL Server Management Studio

1. **Open Windows Settings**: Select the Start menu and select **Settings**.
2. **Navigate to Default Apps**: In the **Settings** window, select **Apps**, then select **Default apps**.
3. **Choose Default Apps by File Type**: Scroll down and Select **Choose default apps by file type**.
4. **Find Your File Extension**: Locate the file extension you want to associate (for example, .sql).
5. **Set SSMS as Default**: Select the current default app and select SQL Server Management Studio from the list.
6. **Configure within SSMS**: Open SSMS, go to **Tools** > **Options**, and under **Text Editor**, specify the editor for the new extension if necessary.

To associate a new file extension with a code editor in SQL Server Management Studio

1. In SQL Server Management Studio, from the **Tools** menu, select **Options**.
2. In the **Options** dialog box, select **Text Editor**, and then select **File Extension**.
3. In the **Extension** box, type your new file extension.
4. In the **Editor** box, select the code editor that you wish to use to open this file type, select **Add**, and then select **OK**.

Related content

- Ssms Utility

ⓘ **Note:** The author created this article with assistance from AI. Learn more

---

Feedback

Was this page helpful? Yes No

Change Font Color, Size, and Style

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can customize the way the text appears in the Editor as well as other places in SQL Server Management Studio.

To change font color, size, and style in the Editor

1. Click **Options** on the **Tools** menu. Click **Environment**, and then click **Fonts and Colors**.
2. In the **Show settings for** list, select **Text Editor**.
3. Change the font, size, display item, foreground and background colors.

> **Note**
>
> Click **Use Defaults** to return to the default settings.

There is only one font setting for the Editor and it affects all editors in SQL Server Management Studio. You can change the color based on the text type (such as comments and statements). Monospace fonts appear in bold in the Font list, and you can apply bold settings on a per-text-type basis. For example, you can format comments and operators bold, and the other text types are unaffected.

The settings for text color and font type are global for all text element types, such as comments, strings, and so on.

See Also

Print Code and Results
Color Coding in Query Editors

Feedback

Was this page helpful? Yes No

Use the Properties Window in Management Studio

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The Properties window describes the state of an item in SQL Server Management Studio, such as a connection or a Showplan operator, and information about database objects such as tables, views, and designers.

You can use the Properties window to view the properties of the current connection. Many properties are read-only in the Properties window but can be changed elsewhere in the Management Studio. For example, the Database property of a query is read-only in the Properties window, but can be changed on the tool bar.

To view properties using the Properties window

1. If the Properties window is not visible, click **Properties Window** on the **View** menu, or press F4.
2. Set the focus on the object that you want to view.
3. Look for a specific property in the Properties window.

To view connection properties of a query window

1. If the Properties window is not visible, click **Properties Window** on the **View** menu, or press F4.
2. In the Properties window, you can see all the connection properties.

To view the properties of a Showplan operator

1. On the **Query** menu, click **Include Actual Execution Plan**.
2. In the SQL Query Editor, type and execute a query.
3. If the Properties window is not visible, click **Properties Window** on the **View** menu, or press F4.

4. On the **Execution plan** tab of the SQL Query Editor click the icons of the operators to view information about the operators in the Properties window.

See Also

Properties Window (Management Studio)

Feedback

Was this page helpful? Yes No

Open an Editor (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes how to open the Database Engine Query, MDX, DMX, or XML/A editors in SQL Server Management Studio. When opened, each editor window appears as a tab in the central pane of Management Studio.

Before You Begin

Management Studio supports four editors: the Database Engine Query Editor for editing Transact-SQL scripts, the DMX and MDX editors for editing scripts using those languages, and the XML/A editor for editing XML/A scripts or XML files. Any of the editors can also be used to edit text files.

Limitations and Restrictions

If you share files with users at other sites that use distinct code pages, you should save your file with the appropriate Unicode code page to prevent errors when reading the file. Also, when saving files for UNIX or Macintosh, be sure to save your files with the appropriate document format. On the **File** menu, click **Save As**, **Save with Encoding** from the down arrow next to the **Save** button, and then choose **Unix** or **Macintosh** under **Line Endings**.

Permissions

Operations you perform in a code editor are subject to the permissions granted to the authentication account you used to log in. For example, if you open a Database Engine Query Editor window using Windows Authentication, you cannot execute Transact-SQL statements that reference objects your Windows login account does not have permissions to access.

How to: Open Editors

This section explains how to open the various editors in SQL Server Management Studio.

Using the File/New Menu

On the **File** menu, click **New**, and then select one of the query editor options:

- **Query with Current Connection** -Opens a new editor window of the type associated with the current connection in Management Studio. The editor window uses the same authentication information as the current connection. For example, if you select an instance of the Database Engine in Object Explorer, and then use **Query with Current Connection**, Management Studio opens a Database Engine Query Editor connected to the same instance using the same authentication information.
- **Database Engine Query** - Opens a new Database Engine Query Editor and a dialog to get the information required to connect to an instance of the Database Engine.
- **Analysis Services MDX Query** - Opens a new Analysis Services MDX Query Editor and a dialog to get the information required to connect to an instance of Analysis Services.
- **Analysis Services DMX Query** - Opens a new Analysis Services DMX Query Editor and a dialog to get the information required to connect to an instance of Analysis Services.
- **Analysis Services XML/A Query** - Opens a new Analysis Services XML/A Query Editor and a dialog to get the information required to connect to an instance of Analysis Services.

Using the File/Open Menu

On the **File** menu, click **Open**, and then navigate to a file and open it. Management Studio opens the appropriate type of editor for the file extension, copies the contents of the file into the editor window, and also opens a connection dialog if needed. For example, if you open a file with a .sql extension, Management Studio opens a Database Engine Query Editor window, copies in the contents of the .sql file, and opens a connection dialog. If you open a file with an extension not associated with a specific editor, Management Studio opens a text editor window and copies in the contents of the file.

For more information, see Associate File Extensions to a Code Editor.

Using the Toolbar

On the **Standard** toolbar, click one of the following buttons:

- **New Query** - -Opens a new editor window of the type associated with the current connection in Management Studio. The editor window uses the same authentication information as the current connection. For example, if you select an instance of the Database Engine in Object Explorer, and then click the **New Query** button, Management Studio opens a Database Engine Query Editor connected to the same instance using the same authentication information.
- **Database Engine Query** - Opens a new Database Engine Query Editor and a dialog to get the information required to connect to an instance of the Database Engine.
- **Analysis Services MDX Query** - Opens a new Analysis Services MDX Query Editor and a dialog to get the information required to connect to an instance of Analysis Services.
- **Analysis Services DMX Query** - Opens a new Analysis Services DMX Query Editor and a dialog to get the information required to connect to an instance of Analysis Services.
- **Analysis Services XML/A Query** - Opens a new Analysis Services XML/A Query Editor and a dialog to get the information required to connect to an instance of Analysis Services.

Using Object Explorer

From **Object Explorer**:

- Right-click the server node connected to an instance of the Database Engine, then select **New Query**. This will open a Database Engine Query Editor window connected to the same instance of the Database Engine and set the database context of the window to the default database for the login.
- Right-click a database node, and then select **New Query**. This will open a Database Engine Query Editor window connected to the same instance of the Database Engine and set the database context of the window to the same database.

Using Solution Explorer

From **Solution Explorer**, expand a folder, right-click an item within the folder, and then click **Open** or double-click the item or file.

Using Template Browser to Open the Database Engine Query Editor

- On the **View** menu, click **Template Explorer**.
- The **Template Browser** window appears in the right pane.
- Double-click a template to open a Database Engine Query window with the text of the template. For example, to open a CREATE DATABASE template, open the **SQL Server Templates** folder, open the **Databases** folder, and double-click **create database**.

Feedback

Was this page helpful? Yes No

Manage the Editor and View Mode

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The Editor gives you a number of ways to control the view of your code.

Changing the View Mode

SQL Server Management Studio features a view mode called **Tabbed Documents**, which allows you to open multiple editors and documents simultaneously and access them through tabs at the top of the Editor. You can alternatively open the Management Studio environment in Multiple Document Interface (MDI) mode, which joins windows without the tabs, and allows each window to be tiled, minimized, and so on.

To switch between view modes

1. Click **Options** on the **Tools** menu.
2. Click **Environment**. Click **General**.
3. Click **Tabbed documents** or **MDI environment**.

> **Note**
>
> The changes will not take effect until SQL Server Management Studio is restarted.

Splitting the View

An Editor window can be split into two separate parts for easier editing.

To split a window

1. Click the splitter bar (located above the scroll bar).
2. Drag the splitter bar downward.
3. To go back to a single pane, double-click the splitter bar dividing the two panes.

The new pane contains the same document, and any changes made to one pane are reflected in the other pane as long as that pane displays the same place in the document.

Word Wrap

When you activate Word Wrap, the horizontal scrollbar is removed and lines of code that exceed the width of the Editor's window size automatically wraps to the next displayed line rather than scrolling off the side of the window.

To activate word wrap

1. Click **Options** on the **Tools** menu.
2. Click **Text Editor**.
3. Open the appropriate language folder (or **All Languages** to affect all languages).
4. Select **Word wrap**.

Enabling Virtual Space Mode

In **Virtual Space** mode, the Editor acts as if the space past the end of each line is filled with an infinite number of spaces, allowing code lines to continue off the side of the visible screen area.

To enable Virtual Space mode

1. Click **Options** on the **Tools** menu.
2. Click **Text Editor**.
3. Open the appropriate language folder (or **All Languages** to affect all languages).
4. Select **Enable virtual space**.

When Virtual Space mode is not enabled, the cursor wraps from the end of one line to the first character of the next line and vice-versa.

Displaying Line Numbers

You can turn on line numbering in your code. Line numbers are useful for navigating code. For more information, see Navigate Code and Text.

> **Note**
>
> Turning on line numbering does not mean that the document will print with line numbers. For line numbers to print, you must select the **Line numbers** check box in the **Page Setup** command on the **File** menu.

To display line numbers in code

1. Click **Options** on the **Tools** menu.
2. Click **Text Editor**.
3. Click **All Languages**.
4. Click **General**.
5. Select **Line numbers**.

To specify line numbering for only some programming languages, select **Line Numbers** in the appropriate folder.

Enabling Full Screen Mode

You can choose to hide all tool windows and view only document windows by enabling Full Screen mode.

To enable Full Screen mode

1. Press ALT+SHIFT+ENTER to toggle Full Screen mode.

Using Auto Hide All

To hide all the tool windows at once

1. Select **Auto Hide All** on the **Window** menu.

Feedback

Was this page helpful? Yes No

Manage Code Formatting

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

With the Editor you can format your code with indenting, hidden text, URLs, and so forth. You can also auto-format your code as you type by using Smart Indenting.

Indenting

You can choose three different styles of text indenting. You can also specify how many spaces compose a single indentation or tab, and whether the Editor uses tabs or space characters when indenting.

To choose an indenting style

1. On the **Tools** menu, click **Options**.
2. Click **Text Editor**.
3. Click the folder, and select **All Languages** to set indenting for all languages.
4. Click **Tabs**.
5. Click one of the following options:
   - **None**. The cursor goes to the beginning of the next line.
   - **Block**. The cursor aligns the next line with the previous line.
   - **Smart** (Default). The language service determines the appropriate indenting style to use.

> **Note**
>
> Some languages do not offer all three indenting options.

To change indent tab settings

1. On the **Tools** menu, click **Options**.

2. Click **Text Editor**.
3. Select the folder for **All Languages** to set indenting for all languages.
4. Click **Tabs**.
5. To specify tab characters for tab and indent operations, click **Keep tabs**. To specify space characters, select **Insert spaces**.

   If you select **Insert Spaces**, enter the number of space characters each tab or indent represents under **Tab Size** or **Indent Size**, respectively.

To indent code

1. Select the text you want to indent.
2. Press TAB, or click the **Indent** button on the Standard toolbar.

To unindent code

1. Select the text you want to unindent.
2. Press SHIFT+TAB, or click the **Unindent** button on the Standard toolbar.

To automatically indent all of your code

1. On the **Tools** menu, click **Options**.
2. Click **Text Editor**.
3. Click **All Languages**.
4. Click **Tabs**.
5. Click **Smart**.

> ⓘ **Note**
>
> The **Smart** option is not available for some languages.

To convert white space to tabs

1. Select the text whose white space you want to convert to tabs.

2. On the **Edit** menu, point to **Advanced**, and click **Tabify Selection**.

To convert tabs to spaces

1. Select the text whose tabs you want to convert to spaces.

2. On the **Edit** menu, point to **Advanced**, and click **Untabify Selection**.

The behavior of these commands depends on the tab settings in the **Options** dialog box. For example, if the tab setting is 4, **Tabify Selection** creates a tab for every 4 contiguous spaces, and **Untabify Selection** creates 4 spaces for every tab.

Converting Text to Upper and Lower Case

You can use commands to convert text to all uppercase or lower case.

To switch text to upper or lower case

1. Select the text you want to convert.

2. To convert text to uppercase, press CTRL+SHIFT+U, or click **Make Uppercase** on the **Advanced** submenu of the **Edit** menu.

3. To convert text to lowercase, press CTRL+SHIFT+L, or click **Make Lowercase** on the **Advanced** submenu of the **Edit** menu.

> **Note**
>
> For a complete list of keyboard shortcut keys, see SQL Server Management Studio Keyboard Shortcuts.

Displaying and Linking to URLs

You can create and display clickable URLs in your code. By default, the URLs:

- Are underlined.
- Change the mouse pointer to a hand when you move over them.
- Open the URL when clicked, if the URL is valid.

To display a clickable URL

1. On the **Tools** menu, click **Options**.
2. Click **Text Editor**.
3. To change the option for only one language, click that language folder and then click **General**. To change the option for all languages, click **All Languages** and then click **General**.
4. Select **Enable single-click URL navigation**.

Feedback

Was this page helpful? Yes No

Navigate Code and Text

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can move through text by using:

- Bookmarks.
- Incremental search.
- Mouse and navigation keys.
- The **Go To** command.

> **Note**
>
> For a complete list of keyboard shortcut keys, see SQL Server Management Studio Keyboard Shortcuts.

Navigating with Bookmarks

To edit a document elsewhere and then return to your original location, add a bookmark. You set bookmarks and move to them using keyboard shortcuts. View the bookmarks in the bookmark window.

Incremental Search

Incremental search helps you navigate directly to locations in the current document as you enter the search characters. You can access incremental search with keyboard shortcuts.

Navigating with the Mouse and Keyboard

The most common way to navigate text is with the mouse and navigation keys:

- Use the LEFT ARROW and RIGHT ARROW keys to move one character at a time or in combination with the CTRL key to move one word at a time. The UP ARROW and DOWN ARROW keys move one line at a time.

- Click a location to place the cursor there.
- Use the scroll bars or scroll wheel on the mouse to move through the text.
- Use the HOME, END, PAGE UP, and PAGE DOWN keys to move through the text.
- Use CTRL+PAGE UP and CTRL+PAGE DOWN to move the insertion point to the top or bottom of the window, respectively.
- Use CTRL+UP ARROW and CTRL+DOWN ARROW to scroll the view without moving the insertion point.

Go To Command

Use the **GoTo** command to go to a specific line number. To make line numbering visible, in the **Options** dialog box, click **Text Editor**, click **All Languages**, click **General**, and then select **Line numbers**.

To go to a specific line number

1. Click **Go To** on the **Edit** menu
2. Enter the line number you want to view

Feedback

Was this page helpful? Yes No

Color Coding in Query Editors

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The text entered in the code editors is assigned a category; each category is identified by a color. The colors help you quickly find text in your code. For example, comments stand out in dark green. The following table lists the most common colors. You can view the whole list of colors and their categories, and configure a custom color scheme by using the **Tools**, **Options** menu. For more information about how to change the default colors, see Change Font Color, Size, and Style.

Default Code Colors

Expand table

Color	Category
Red	SQL string
Dark green	Comment
Black on silver background	SQLCMD command
Magenta	System function
Green	System table, view, or table-valued function. Also, the system schemas sys and INFORMATION_SCHEMA.
Blue	Keyword
Teal	Line numbers or template parameter
Maroon	SQL Server stored procedure
Dark gray	Operators

Status Bar

You can configure registered servers or Database Engine servers in Object Explorer to have different colors in the Database Engine Query Editor status bar. This helps you identify which server each editor window is connected to when you have many windows

open at the same time. For more information about setting status bar colors, see Status Bar (Database Engine Query Editor).

Some types of editors do not display the status bar, or do not support multiple colors.

Feedback

Was this page helpful? Yes No

Configure Editors (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can customize the operation of the SQL Server Management Studio editors by configuring the options for each editor.

Setting Editor Options

Most of the editor options are set by using the **Tools** menu and selecting **Options...** to display an **Options** dialog. In the **Options** dialog, open the **Text Editor** node in the left pane to set code and text editing options. The nodes under Text Editor apply to specific editors:

1. **All Languages** - options set using this node apply to all of the Management Studio editors. You can override these settings by using the other nodes to set different options for a specific editor.
2. **Plain Text** - options set using this node apply to the MDX, DMX, and text editors.
3. **Transact-SQL** - options set using this node apply to the Database Engine Query Editor.
4. **XML** - options set using this node apply to the XML for Analysis editor.

Open the **Query Execution** or **Query Results** nodes to customize the execution of queries and how the results are displayed.

Editor Configuration Tasks

Expand table

Task Description	Topic
Describes how to specify that an editor be opened by double-clicking on a file with a specified extension in Windows Explorer.	Associate File Extensions to a Code Editor
Describes how to customize fonts to make code and text more	Change Font Color, Size, and

Task Description	Topic
readable.	Style
Describes how to view properties.	Use the Properties Window in Management Studio
Location of the F1 help pages for the editor options dialogs.	Query Options Pages F1 Help

Feedback

Was this page helpful? Yes No

Associate file extensions to a code editor

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Associating file extensions to a specific code editor allows you to open a file with the appropriate SQL Server Management Studio code editor when you double-click a file in Windows Explorer. The extensions commonly used by Management Studio, such as .sql and .mdx, are associated during installation. New file extensions must also be associated to Management Studio in the file system. You can use this feature to open files created with other editors, or to open files you have renamed, such as backups of .sql files that were renamed .bak.

There are two steps in the process. First associate the extension with Management Studio, and then associate the extension with a specific code editor.

To associate a new file extension with SQL Server Management Studio

1. **Open Windows Settings**: Select the Start menu and select **Settings**.
2. **Navigate to Default Apps**: In the **Settings** window, select **Apps**, then select **Default apps**.
3. **Choose Default Apps by File Type**: Scroll down and Select **Choose default apps by file type**.
4. **Find Your File Extension**: Locate the file extension you want to associate (for example, .sql).
5. **Set SSMS as Default**: Select the current default app and select SQL Server Management Studio from the list.
6. **Configure within SSMS**: Open SSMS, go to **Tools** > **Options**, and under **Text Editor**, specify the editor for the new extension if necessary.

To associate a new file extension with a code editor in SQL Server Management Studio

1. In SQL Server Management Studio, from the **Tools** menu, select **Options**.
2. In the **Options** dialog box, select **Text Editor**, and then select **File Extension**.
3. In the **Extension** box, type your new file extension.
4. In the **Editor** box, select the code editor that you wish to use to open this file type, select **Add**, and then select **OK**.

Related content

- Ssms Utility

ⓘ **Note:** The author created this article with assistance from AI. Learn more

---

Feedback

Was this page helpful? Yes No

Change Font Color, Size, and Style

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can customize the way the text appears in the Editor as well as other places in SQL Server Management Studio.

To change font color, size, and style in the Editor

1. Click **Options** on the **Tools** menu. Click **Environment**, and then click **Fonts and Colors**.
2. In the **Show settings for** list, select **Text Editor**.
3. Change the font, size, display item, foreground and background colors.

> **Note**
>
> Click **Use Defaults** to return to the default settings.

There is only one font setting for the Editor and it affects all editors in SQL Server Management Studio. You can change the color based on the text type (such as comments and statements). Monospace fonts appear in bold in the Font list, and you can apply bold settings on a per-text-type basis. For example, you can format comments and operators bold, and the other text types are unaffected.

The settings for text color and font type are global for all text element types, such as comments, strings, and so on.

See Also

Print Code and Results
Color Coding in Query Editors

Feedback

Was this page helpful? Yes No

Use the Properties Window in Management Studio

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The Properties window describes the state of an item in SQL Server Management Studio, such as a connection or a Showplan operator, and information about database objects such as tables, views, and designers.

You can use the Properties window to view the properties of the current connection. Many properties are read-only in the Properties window but can be changed elsewhere in the Management Studio. For example, the Database property of a query is read-only in the Properties window, but can be changed on the tool bar.

To view properties using the Properties window

1. If the Properties window is not visible, click **Properties Window** on the **View** menu, or press F4.
2. Set the focus on the object that you want to view.
3. Look for a specific property in the Properties window.

To view connection properties of a query window

1. If the Properties window is not visible, click **Properties Window** on the **View** menu, or press F4.
2. In the Properties window, you can see all the connection properties.

To view the properties of a Showplan operator

1. On the **Query** menu, click **Include Actual Execution Plan**.
2. In the SQL Query Editor, type and execute a query.
3. If the Properties window is not visible, click **Properties Window** on the **View** menu, or press F4.

4. On the **Execution plan** tab of the SQL Query Editor click the icons of the operators to view information about the operators in the Properties window.

See Also

Properties Window (Management Studio)

Feedback

Was this page helpful? Yes No

Open an Editor (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes how to open the Database Engine Query, MDX, DMX, or XML/A editors in SQL Server Management Studio. When opened, each editor window appears as a tab in the central pane of Management Studio.

Before You Begin

Management Studio supports four editors: the Database Engine Query Editor for editing Transact-SQL scripts, the DMX and MDX editors for editing scripts using those languages, and the XML/A editor for editing XML/A scripts or XML files. Any of the editors can also be used to edit text files.

Limitations and Restrictions

If you share files with users at other sites that use distinct code pages, you should save your file with the appropriate Unicode code page to prevent errors when reading the file. Also, when saving files for UNIX or Macintosh, be sure to save your files with the appropriate document format. On the **File** menu, click **Save As**, **Save with Encoding** from the down arrow next to the **Save** button, and then choose **Unix** or **Macintosh** under **Line Endings**.

Permissions

Operations you perform in a code editor are subject to the permissions granted to the authentication account you used to log in. For example, if you open a Database Engine Query Editor window using Windows Authentication, you cannot execute Transact-SQL statements that reference objects your Windows login account does not have permissions to access.

How to: Open Editors

This section explains how to open the various editors in SQL Server Management Studio.

Using the File/New Menu

On the **File** menu, click **New**, and then select one of the query editor options:

- **Query with Current Connection** -Opens a new editor window of the type associated with the current connection in Management Studio. The editor window uses the same authentication information as the current connection. For example, if you select an instance of the Database Engine in Object Explorer, and then use **Query with Current Connection**, Management Studio opens a Database Engine Query Editor connected to the same instance using the same authentication information.
- **Database Engine Query** - Opens a new Database Engine Query Editor and a dialog to get the information required to connect to an instance of the Database Engine.
- **Analysis Services MDX Query** - Opens a new Analysis Services MDX Query Editor and a dialog to get the information required to connect to an instance of Analysis Services.
- **Analysis Services DMX Query** - Opens a new Analysis Services DMX Query Editor and a dialog to get the information required to connect to an instance of Analysis Services.
- **Analysis Services XML/A Query** - Opens a new Analysis Services XML/A Query Editor and a dialog to get the information required to connect to an instance of Analysis Services.

Using the File/Open Menu

On the **File** menu, click **Open**, and then navigate to a file and open it. Management Studio opens the appropriate type of editor for the file extension, copies the contents of the file into the editor window, and also opens a connection dialog if needed. For example, if you open a file with a .sql extension, Management Studio opens a Database Engine Query Editor window, copies in the contents of the .sql file, and opens a connection dialog. If you open a file with an extension not associated with a specific editor, Management Studio opens a text editor window and copies in the contents of the file.

For more information, see Associate File Extensions to a Code Editor.

Using the Toolbar

On the **Standard** toolbar, click one of the following buttons:

- **New Query** - -Opens a new editor window of the type associated with the current connection in Management Studio. The editor window uses the same authentication information as the current connection. For example, if you select an instance of the Database Engine in Object Explorer, and then click the **New Query** button, Management Studio opens a Database Engine Query Editor connected to the same instance using the same authentication information.
- **Database Engine Query** - Opens a new Database Engine Query Editor and a dialog to get the information required to connect to an instance of the Database Engine.
- **Analysis Services MDX Query** - Opens a new Analysis Services MDX Query Editor and a dialog to get the information required to connect to an instance of Analysis Services.
- **Analysis Services DMX Query** - Opens a new Analysis Services DMX Query Editor and a dialog to get the information required to connect to an instance of Analysis Services.
- **Analysis Services XML/A Query** - Opens a new Analysis Services XML/A Query Editor and a dialog to get the information required to connect to an instance of Analysis Services.

Using Object Explorer

From **Object Explorer**:

- Right-click the server node connected to an instance of the Database Engine, then select **New Query**. This will open a Database Engine Query Editor window connected to the same instance of the Database Engine and set the database context of the window to the default database for the login.
- Right-click a database node, and then select **New Query**. This will open a Database Engine Query Editor window connected to the same instance of the Database Engine and set the database context of the window to the same database.

Using Solution Explorer

From **Solution Explorer**, expand a folder, right-click an item within the folder, and then click **Open** or double-click the item or file.

Using Template Browser to Open the Database Engine Query Editor

- On the **View** menu, click **Template Explorer**.
- The **Template Browser** window appears in the right pane.
- Double-click a template to open a Database Engine Query window with the text of the template. For example, to open a CREATE DATABASE template, open the **SQL Server Templates** folder, open the **Databases** folder, and double-click **create database**.

Feedback

Was this page helpful? Yes No

Manage the Editor and View Mode

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The Editor gives you a number of ways to control the view of your code.

Changing the View Mode

SQL Server Management Studio features a view mode called **Tabbed Documents**, which allows you to open multiple editors and documents simultaneously and access them through tabs at the top of the Editor. You can alternatively open the Management Studio environment in Multiple Document Interface (MDI) mode, which joins windows without the tabs, and allows each window to be tiled, minimized, and so on.

To switch between view modes

1. Click **Options** on the **Tools** menu.
2. Click **Environment**. Click **General**.
3. Click **Tabbed documents** or **MDI environment**.

> **Note**
>
> The changes will not take effect until SQL Server Management Studio is restarted.

Splitting the View

An Editor window can be split into two separate parts for easier editing.

To split a window

1. Click the splitter bar (located above the scroll bar).
2. Drag the splitter bar downward.
3. To go back to a single pane, double-click the splitter bar dividing the two panes.

The new pane contains the same document, and any changes made to one pane are reflected in the other pane as long as that pane displays the same place in the document.

Word Wrap

When you activate Word Wrap, the horizontal scrollbar is removed and lines of code that exceed the width of the Editor's window size automatically wraps to the next displayed line rather than scrolling off the side of the window.

To activate word wrap

1. Click **Options** on the **Tools** menu.
2. Click **Text Editor**.
3. Open the appropriate language folder (or **All Languages** to affect all languages).
4. Select **Word wrap**.

Enabling Virtual Space Mode

In **Virtual Space** mode, the Editor acts as if the space past the end of each line is filled with an infinite number of spaces, allowing code lines to continue off the side of the visible screen area.

To enable Virtual Space mode

1. Click **Options** on the **Tools** menu.
2. Click **Text Editor**.
3. Open the appropriate language folder (or **All Languages** to affect all languages).
4. Select **Enable virtual space**.

When Virtual Space mode is not enabled, the cursor wraps from the end of one line to the first character of the next line and vice-versa.

Displaying Line Numbers

You can turn on line numbering in your code. Line numbers are useful for navigating code. For more information, see Navigate Code and Text.

> **Note**
>
> Turning on line numbering does not mean that the document will print with line numbers. For line numbers to print, you must select the **Line numbers** check box in the **Page Setup** command on the **File** menu.

To display line numbers in code

1. Click **Options** on the **Tools** menu.
2. Click **Text Editor**.
3. Click **All Languages**.
4. Click **General**.
5. Select **Line numbers**.

To specify line numbering for only some programming languages, select **Line Numbers** in the appropriate folder.

Enabling Full Screen Mode

You can choose to hide all tool windows and view only document windows by enabling Full Screen mode.

To enable Full Screen mode

1. Press ALT+SHIFT+ENTER to toggle Full Screen mode.

Using Auto Hide All

To hide all the tool windows at once

1. Select **Auto Hide All** on the **Window** menu.

Feedback

Was this page helpful? Yes No

Manage Code Formatting

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

With the Editor you can format your code with indenting, hidden text, URLs, and so forth. You can also auto-format your code as you type by using Smart Indenting.

Indenting

You can choose three different styles of text indenting. You can also specify how many spaces compose a single indentation or tab, and whether the Editor uses tabs or space characters when indenting.

To choose an indenting style

1. On the **Tools** menu, click **Options**.
2. Click **Text Editor**.
3. Click the folder, and select **All Languages** to set indenting for all languages.
4. Click **Tabs**.
5. Click one of the following options:
   - **None**. The cursor goes to the beginning of the next line.
   - **Block**. The cursor aligns the next line with the previous line.
   - **Smart** (Default). The language service determines the appropriate indenting style to use.

> **Note**
>
> Some languages do not offer all three indenting options.

To change indent tab settings

1. On the **Tools** menu, click **Options**.

2. Click **Text Editor**.
3. Select the folder for **All Languages** to set indenting for all languages.
4. Click **Tabs**.
5. To specify tab characters for tab and indent operations, click **Keep tabs**. To specify space characters, select **Insert spaces**.

   If you select **Insert Spaces**, enter the number of space characters each tab or indent represents under **Tab Size** or **Indent Size**, respectively.

To indent code

1. Select the text you want to indent.
2. Press TAB, or click the **Indent** button on the Standard toolbar.

To unindent code

1. Select the text you want to unindent.
2. Press SHIFT+TAB, or click the **Unindent** button on the Standard toolbar.

To automatically indent all of your code

1. On the **Tools** menu, click **Options**.
2. Click **Text Editor**.
3. Click **All Languages**.
4. Click **Tabs**.
5. Click **Smart**.

> ⓘ **Note**
>
> The **Smart** option is not available for some languages.

To convert white space to tabs

1. Select the text whose white space you want to convert to tabs.

2. On the **Edit** menu, point to **Advanced**, and click **Tabify Selection**.

To convert tabs to spaces

1. Select the text whose tabs you want to convert to spaces.

2. On the **Edit** menu, point to **Advanced**, and click **Untabify Selection**.

The behavior of these commands depends on the tab settings in the **Options** dialog box. For example, if the tab setting is 4, **Tabify Selection** creates a tab for every 4 contiguous spaces, and **Untabify Selection** creates 4 spaces for every tab.

Converting Text to Upper and Lower Case

You can use commands to convert text to all uppercase or lower case.

To switch text to upper or lower case

1. Select the text you want to convert.

2. To convert text to uppercase, press CTRL+SHIFT+U, or click **Make Uppercase** on the **Advanced** submenu of the **Edit** menu.

3. To convert text to lowercase, press CTRL+SHIFT+L, or click **Make Lowercase** on the **Advanced** submenu of the **Edit** menu.

> **Note**
>
> For a complete list of keyboard shortcut keys, see SQL Server Management Studio Keyboard Shortcuts.

Displaying and Linking to URLs

You can create and display clickable URLs in your code. By default, the URLs:

- Are underlined.
- Change the mouse pointer to a hand when you move over them.
- Open the URL when clicked, if the URL is valid.

To display a clickable URL

1. On the **Tools** menu, click **Options**.
2. Click **Text Editor**.
3. To change the option for only one language, click that language folder and then click **General**. To change the option for all languages, click **All Languages** and then click **General**.
4. Select **Enable single-click URL navigation**.

Feedback

Was this page helpful? Yes No

Navigate Code and Text

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can move through text by using:

- Bookmarks.
- Incremental search.
- Mouse and navigation keys.
- The **Go To** command.

> **Note**
>
> For a complete list of keyboard shortcut keys, see SQL Server Management Studio Keyboard Shortcuts.

Navigating with Bookmarks

To edit a document elsewhere and then return to your original location, add a bookmark. You set bookmarks and move to them using keyboard shortcuts. View the bookmarks in the bookmark window.

Incremental Search

Incremental search helps you navigate directly to locations in the current document as you enter the search characters. You can access incremental search with keyboard shortcuts.

Navigating with the Mouse and Keyboard

The most common way to navigate text is with the mouse and navigation keys:

- Use the LEFT ARROW and RIGHT ARROW keys to move one character at a time or in combination with the CTRL key to move one word at a time. The UP ARROW and DOWN ARROW keys move one line at a time.

- Click a location to place the cursor there.
- Use the scroll bars or scroll wheel on the mouse to move through the text.
- Use the HOME, END, PAGE UP, and PAGE DOWN keys to move through the text.
- Use CTRL+PAGE UP and CTRL+PAGE DOWN to move the insertion point to the top or bottom of the window, respectively.
- Use CTRL+UP ARROW and CTRL+DOWN ARROW to scroll the view without moving the insertion point.

Go To Command

Use the **GoTo** command to go to a specific line number. To make line numbering visible, in the **Options** dialog box, click **Text Editor**, click **All Languages**, click **General**, and then select **Line numbers**.

To go to a specific line number

1. Click **Go To** on the **Edit** menu
2. Enter the line number you want to view

Feedback

Was this page helpful? Yes No

SQL Server Management Objects (SMO) Programming Guide

Article • 11/22/2024

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics SQL database in Microsoft Fabric

SQL Server Management Objects (SMO) is a collection of objects that are designed for programming all aspects of managing Microsoft SQL Server. SQL Server Replication Management Objects (RMO) is a collection of objects that encapsulates SQL Server replication management.

Expand table

Topic	Description
Getting Started in SMO	Provides information on how to start developing a SMO application
Creating SMO Programs Programming Specific Tasks	Provides information about programming the SMO objects in the Microsoft.SqlServer.management, Microsoft.SqlServer.Management.NotificationServices, Microsoft.SqlServer.Management.Smo, Microsoft.SqlServer.Management.Smo.Agent, Microsoft.SqlServer.Management.Smo.Broker, Microsoft.SqlServer.Management.Smo.Mail, Microsoft.SqlServer.Management.Smo.RegisteredServers, Microsoft.SqlServer.Management.Smo.Wmi, and Microsoft.SqlServer.Management.Trace namespaces. This includes instructions to write programs that define databases and manage SQL Server. You can use SMO to create databases, perform backups, create jobs, configure SQL Server, assign permissions, and to perform many other administrative tasks.
Replication Developer Documentation	Provides information about programming the RMO objects in the Microsoft.SqlServer.Replication namespace.

See Also

Replication Developer Documentation

The SQL Server Management Studio Environment

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

SQL Server Management Studio provides a rich environment for managing and developing queries in SQL Server.

> **Note**
>
> Conflicts with graphics drivers have been found to cause issues in the SQL Server Management Studio rendering process and as such hardware graphics acceleration is disabled by default. If you experience rendering or other visual problems in the SSMS interface, please verify "Use hardware graphics acceleration if available" is *unchecked* in Tools>Options.

In This Section

General User Interface Elements
Outline general features of the user interface, such as Solution Explorer, Object Explorer, the Properties window, and the integrated Web browser.

Solution Explorer
Introduces Solution Explorer, which provides you with an organized view of your projects and their files as well as ready access to commands for managing them.

Solution Explorer Source Control
Describes how SQL Server Management Studio integrates with source control products.

Feedback

Was this page helpful? Yes No

General User Interface Elements

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The **Editor**, the **Properties** window, and the **Toolbox** are some of the basic elements of Microsoft SQL Server Management Studio.

In This Section

SQL Server Management Studio Web Browser
Describes how SQL Server Management Studio invokes a web browser.

External Tools Dialog Box
Allows you to add tools to the **Tools** menu that are not part of SQL Server Management Studio.

Properties Window (Management Studio)
Allows you to view properties and events of selected objects that are located in editors and designers. Also allows you to edit and view the properties of files, projects, and solutions. Some properties can be edited in the Properties window.

Use the Toolbox
Explains how to manipulate and use the **Toolbox** to add controls to your project's designers. For example, you use the **Toolbox** to add steps to a Maintenance task.

Feedback

Was this page helpful? Yes No

SQL Server Management Studio Web Browser

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

SQL Server Management Studio (SSMS) can invoke a web browser through menu items, when selecting a URL in the query editor, or when using Microsoft Entra authentication.

Web browser configuration

To control what browser is invoked from SSMS, change the setting, **Use system default web browser**. Within **Tools** > **Options**, select **Azure Services**, and within **Miscellaneous**, alter the setting.

- The default value for **Use system default web browser** is *False* starting with SSMS 19.1. When set to *False*, SSMS uses the default browser configured for the user's workstation. This value isn't changed when upgrading from an earlier version of SSMS to version 19.1 or higher.
- For SSMS 19.0.2 and below, the default value is *True*. When set to *True*, SSMS invokes Microsoft Internet Explorer for Microsoft Entra authentication. Internet Explorer was retired in June 2022. If you encounter an error message titled `Unsupported browser`, change **Use system default web browser** to *False* and configure the default browser for the workstation.

See Also

- General User Interface Elements
- Options (Azure Services)

Feedback

Was this page helpful? Yes No

Property Pages in SQL Server Management Studio

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Property page dialog boxes in Microsoft SQL Server Management Studio all use a common format displaying information with expanding and collapsing categories. The fields shown depend on the particular property. Properties shown in gray are read-only. Categorized and Alphabetic buttons are near the top of each property page.

The following table describes the common elements of Management Studio property page dialog boxes.

Expand table

Element	Description
Categorized	Lists all properties and property values for the selected object, sorted by category. In category view, you can collapse a category to reduce the number of visible properties. When you expand or collapse a category, you see a plus sign (+) or minus sign (-) to the left of the category name. Categories are listed alphabetically.
Alphabetic	Lists all properties and property values for the selected object, sorted alphabetically.
Property name	The first column in the grid lists the property names.
Properties	The second column in the grid lists the property values.
Description pane	The description pane appears at the bottom of the page and shows the property type and a short description of the property. You can turn the description of the property off and on using the **Description** command on the shortcut menu.

Feedback

Was this page helpful? Yes No

Properties Window (Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use this window to view properties of selected elements. You can also use the Properties window to view file, project, and solution properties. The Properties window is available by clicking **Properties Window** on the **View** menu.

The Properties window displays different types of editing fields, depending on the needs of a particular property. Properties shown in gray are read-only.

Options

Expand table

Element	Description
Object name	Lists the currently selected object or objects. Only objects from the active editor or designer are visible.
Categorized	Lists all properties and property values for the selected object, by category. You can collapse a category to reduce the number of visible properties. When you expand or collapse a category, you see a plus (+) or minus (-) to the left of the category name. Categories are listed alphabetically.
Alphabetic	Alphabetically sorts all design-time properties and events for selected objects.
Properties	Displays the properties for an object.
Description pane	The description pane appears at the bottom of the Properties window and shows the property type and a short description of the property. You can turn the description of the property off and on using the **Description** command on the shortcut menu.

See Also

General User Interface Elements

Feedback

Was this page helpful? Yes No

Use the Toolbox

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can select and drag items, text, and controls from the Toolbox onto forms, pages, and designers, and drag items from these sources back into the Toolbox for reuse later. For example, in SQL Server Management Studio you can drag a task onto a Maintenance Plan.

Items in the Toolbox are grouped into sections called tabs. Here is a reference list of common Toolbox tasks and how to do them:

Expand table

To	Do this
Open the Toolbox	On the **View** menu, click **Toolbox**.
Make the Toolbox close automatically	Open the Toolbox. On the **Window** menu, select **Auto Hide**.
Make the Toolbox stay open	Open the Toolbox. On the **Window** menu, clear **Auto Hide**.
Move the Toolbox to a different location	Open the Toolbox. On the **Window** menu, clear **Auto Hide**, and then select **Floating**. Drag the Toolbox to the desired location.
Hide the Toolbox	In the **Window** menu, select **Hide**. (To reopen the Toolbox, click **Toolbox** on the **View** menu.)
Expand a Toolbox tab	Click the desired tab in the Toolbox.
Expand Toolbox tabs one after another	Press CTRL+DOWN ARROW to expand the next Toolbox tab, or CTRL+UP ARROW to expand the previous tab.
Create a new Toolbox tab	Right-click anywhere in the Toolbox, and then click **Add Tab**. Type the name for the new tab, and then press Enter.
Insert a Toolbox item at the selected location on the designer	Drag an item from the Toolbox to the designer or double-click the desired item on the expanded Toolbox tab.
Change the position of a tab in the Toolbox	Drag the Toolbox tab to the new location preferred, and release the mouse.
Change the position of an item on an expanded Toolbox tab	Drag the item to the new location, and release the mouse.

To	Do this
Rename a Toolbox item	Right-click the Toolbox item, and click **Rename Item** on the shortcut menu.

See Also

General User Interface Elements

Feedback

Was this page helpful? Yes No

F1 Help for Server Connections (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This section contains the F1 Help for the **Connect to Server** dialog box pages in SQL Server Management Studio.

- SQL Server Browser service
- Connect to Server (Analysis Services)
- Connect to Server (Connection Properties page) - Analysis Services
- Connect to Server (Connection Properties page) - Database Engine
- Integration Services service (SSIS service)
- Connect to a Report Server in Management Studio
- Connect to Server (Database Engine)
- Integration Services Service (SSIS Service)
- Connect to Server (Login page) - Analysis Services
- Connect to Server (Login page) - Database Engine
- Connect to Server (Always Encrypted page) - Database Engine
- Integration Services Service (SSIS Service)
- Connect to a Report Server in Management Studio
- Choose an authentication mode

Feedback

Was this page helpful? Yes No

SQL Server Management Studio (SSMS) Query Editor

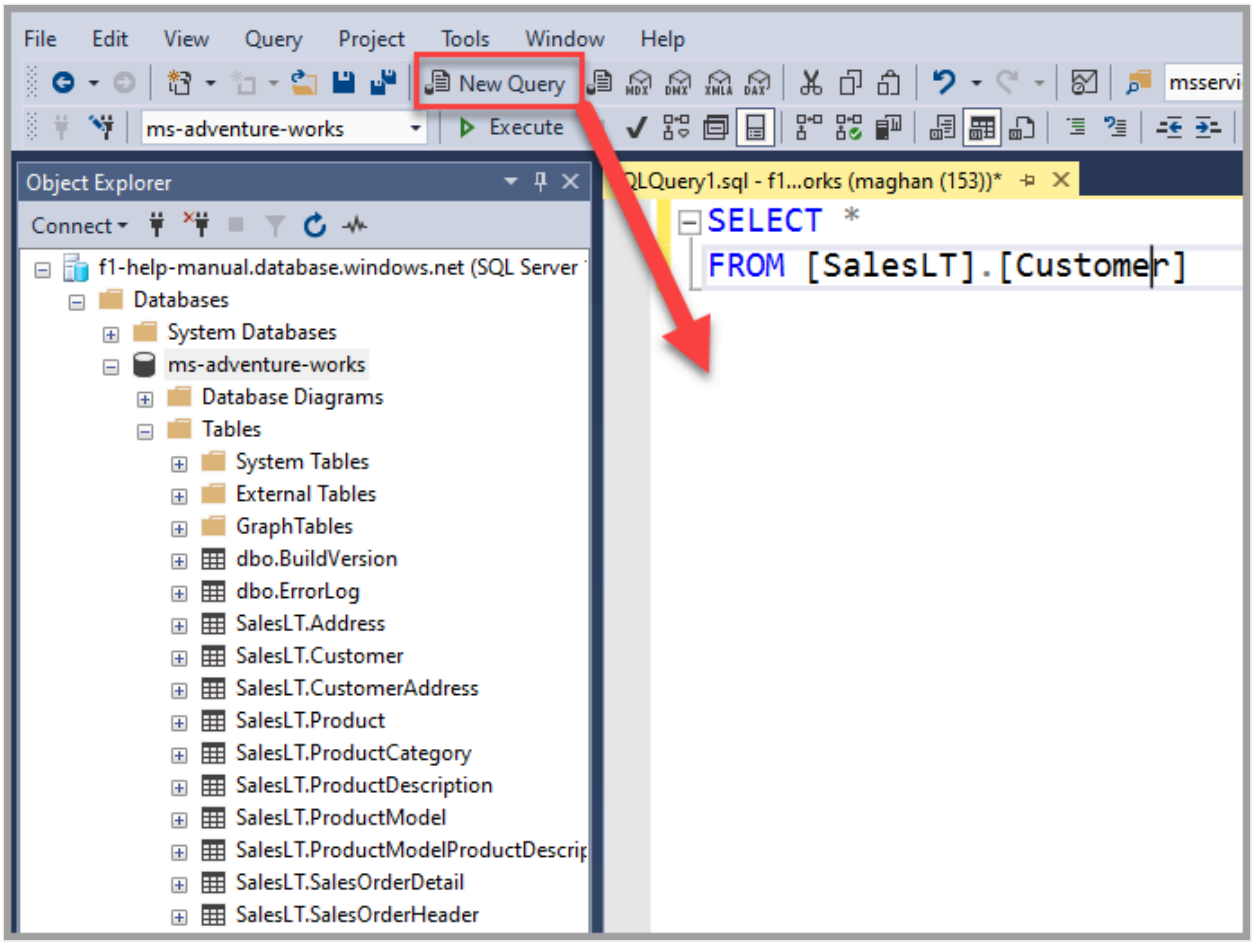
Article • 02/13/2025

Applies to: SQL Server · Azure SQL Database · Azure SQL Managed Instance · Azure Synapse Analytics

This article explains the query editor's features and functions in SQL Server Management Studio (SSMS).

> **Note**
>
> If you want to learn how to use Transact-SQL (T-SQL) F1 Help, please view the Transact-SQL F1 Help section.
>
> If you want to learn about tasks that you can do with the editor, please visit the Editor tasks section.

The editors in SSMS share a typical architecture. The text editor implements the base level of functionality and can be used as a basic editor for text files. The other editors, or query editors, extend this functionality base by including a language service that defines the syntax of one of the languages supported in SQL Server. The query editors also implement varying levels of support for editor features such as IntelliSense and debugging. The query editors include the Database Engine Query Editor for use in building scripts containing T-SQL and XQuery statements, the MDX editor for the MDX language, the DMX editor for the DMX language, and the XML/A editor for the XML for Analysis language. You can use the Query Editor to create and run scripts containing Transact-SQL statements.



SQL Editor toolbar

When the Query Editor is open, the SQL Editor toolbar appears with the following buttons.

You can also add the SQL Editor toolbar by selecting the **View** menu, selecting **Toolbars**, and then selecting **SQL Editor**. If you add the SQL Editor toolbar when no Query Editor windows are open, all the buttons are unavailable.

Connect using the editor toolbar

Opens the Connect to Server dialog box. Use this dialog box to establish a connection to a server.

You can also connect to your database using the context menu.

Change connection using the editor toolbar

Opens the **Connect to Server** dialog box. Use this dialog box to establish a connection to a different server.

You can also change connections using the context menu.

Available databases using the editor toolbar

Change the connection to a different database on the same server.

Execute using the editor toolbar

Executes the selected code or, if no code is selected, it executes all the Query Editor code.

You can also **Execute** a query by selecting F5 or from the context menu.

Cancel executing query using the editor toolbar

Sends a cancellation request to the server. Some queries can't be canceled immediately but must wait for a suitable cancellation condition. When transactions are canceled, delays might occur while transactions are rolled back.

You can also cancel an executing query by selecting Alt + Break.

Parse using the editor toolbar

Checks the syntax of the selected code. If no code is selected, it checks all the code's syntax in the Query Editor window.

You can also check the code in the Query Editor by selecting Ctrl + F5.

Display estimated execution plan using the editor toolbar

Requests a query execution plan from the query processor without executing the query, and displays the plan in the **Execution plan** window. This plan uses index statistics to estimate the number of rows expected to return during each part of the query execution. The actual query plan that is used can be different from the estimated execution plan. If the number of returned rows is different from the estimate, and the query processor changes the plan to be more efficient, this can occur.

You can also display an estimated execution plan by selecting Ctrl + L or from the context menu.

Query options using the editor toolbar

Opens the **Query Options** dialog box. Use this dialog box to configure the default options for query execution and for query results.

You can also select **Query Options** from the context menu.

IntelliSense enabled using the editor toolbar

Specifies whether IntelliSense functionality is available in the Database Engine Query Editor. This option is set by default.

You can also select **IntelliSense Enabled** by selecting Ctrl + B then Ctrl + I, or from the context menu.

Include actual execution plan using the editor toolbar

Executes the query, returns the query results, and uses the execution plan for the query. The queries appear as a graphical query plan in the **Execution plan** window.

You can also select the **Include Actual Execution Plan** by selecting Ctrl + M or from the context menu.

Include live query statistics using the editor toolbar

Provides real-time insights into the query execution process as the controls flow from one query plan operator to another.

You can also select the **Include Live Query Statistics** from the context menu.

Include client statistics using the editor toolbar

Includes a **Client Statistics** window that contains statistics about the query and about the network packets, and the elapsed time of the query.

You can also select the **Include Live Query Statistics** by selecting Shift + Alt + S or from the context menu.

Results to text using the editor toolbar

Returns the query results as text in the **Results** window.

You can also return results to text by selecting Ctrl + T or from the context menu.

Results to grid using the editor toolbar

Returns the query results as one or more grids in the **Results** window. This option is enabled by default.

You can also return results to text by selecting Ctrl + D or from the context menu.

Results to file using the editor toolbar

When the query executes, the **Save Results** dialog box opens. In **Save In**, select the folder in which you want to save the file. In **Filename**, type the name of the file, and then select **Save** to save the query results as a **Report** file that has the .rpt extension. For advanced options, select the down-arrow on the **Save** button, and then select **Save with Encoding**.

You can also return results to text by selecting Ctrl + Shift + F or from the context menu.

Comment out the selected lines using the editor toolbar

Makes the current line a comment by adding a comment operator (--) at the beginning of the line.

You can also comment out a line by selecting Ctrl + K, then Ctrl + C.

Uncomment the selected lines using the editor toolbar

Makes the current line an active source statement by removing any comment operator (--) at the beginning of the line.

You can also uncomment a line by selecting Ctrl + K, then Ctrl + U.

Decrease indent using the editor toolbar

Moves the text of the line to the left by removing blanks at the beginning of the line.

Increase line indent using the editor toolbar

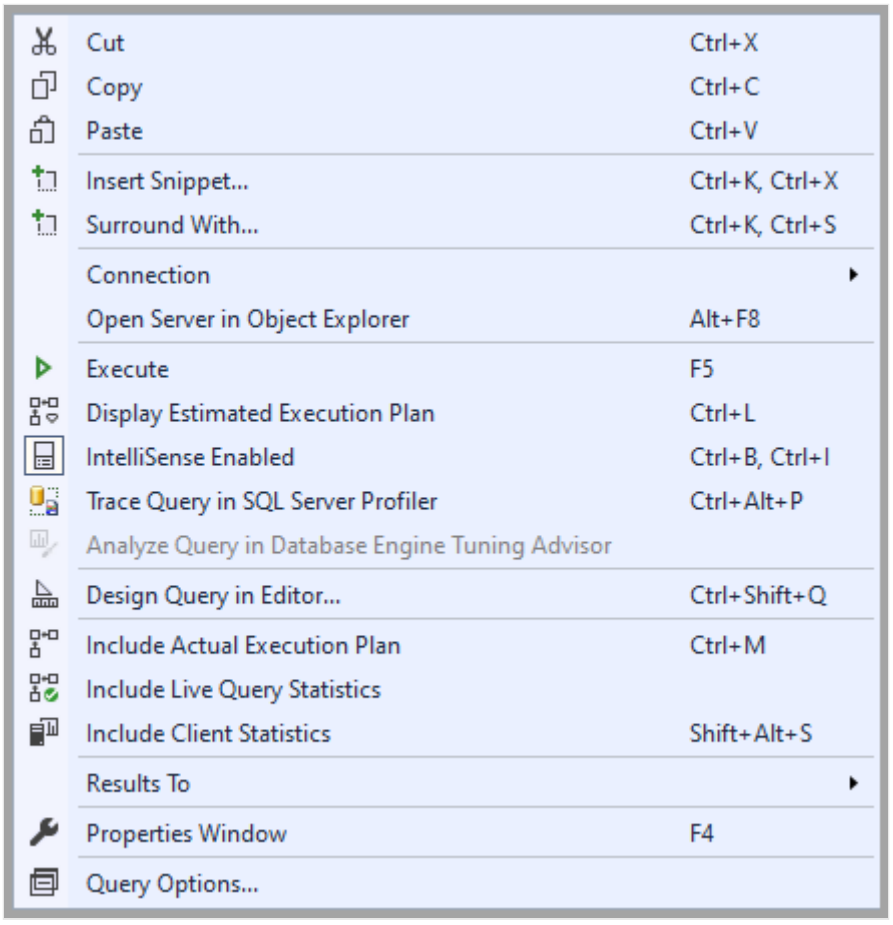
Moves the text of the line to the right by adding blanks at the beginning of the line.

Specify values for template parameters using the editor toolbar

Opens a dialog box that you can use to specify values for parameters in stored procedures and functions.

Context menu

You can access the context menu by *right-clicking* anywhere in the query editor. The options in the context menu are similar to the SQL Editor Toolbar. With the context menu, you see the same options as **Connect** and **Execute**, but you also get other options listed, such as **Insert Snippet** and **Surround With**.



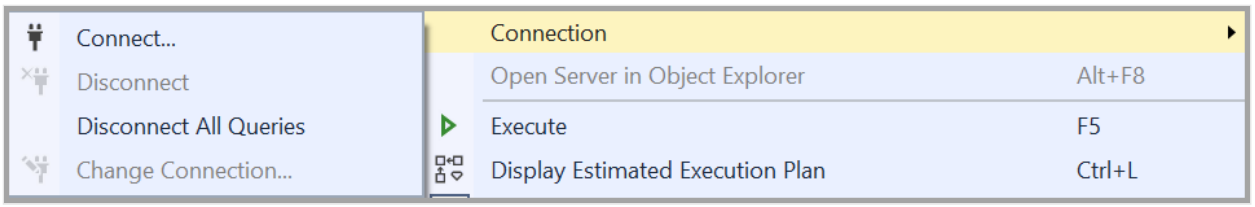
Insert snippet using the context menu

Transact-SQL Snippets are templates you can use as a starting point when writing new Transact-SQL statements in the Query Editor.

Surround-with using the context menu

A surround-with snippet is a template you can use as a starting point when enclosing a set of Transact-SQL statements in a BEGIN, IF, or WHILE block.

Connection using the context menu



There are more **Connection** options in the context menu compared to the toolbar options in SSMS.

- **Connect** - Opens the Connect to Server dialog box. Use this dialog box to establish a connection to a server.
- **Disconnect** - Disconnects the current Query Editor from the server.
- **Disconnect All Queries** - Disconnects all query connections.
- **Change Connection** - Opens the Connect to Server dialog box. Use this dialog box to establish a connection to a different server.

Open server in Object Explorer using the context menu

Object Explorer provides a hierarchical user interface to view and manage the objects in each instance of SQL Server. The Object Explorer Details pane presents a tabular view of instance objects, and the capability to search for specific objects. The capabilities of Object Explorer vary slightly depending on the type of server, but generally include the development features for databases, and management features for all server types.

Execute using the context menu

Executes the selected code or, if no code is selected, executes all the code in the Query Editor.

Display estimated execution plan using the context menu

Requests a query execution plan from the query processor without actually executing the query, and displays the plan in the **Execution plan** window. This plan uses index statistics to estimate the number of rows expected to return during each part of the query execution. The actual query plan that is used can be different from the estimated

execution plan. If the number of rows that are returned is different from the estimate, and the query processor changes the plan to be more efficient, this can occur.

IntelliSense enabled using the context menu

Specifies whether IntelliSense functionality is available in the Database Engine Query Editor. This option is set by default.

Trace query in SQL Server Profiler using the context menu

SQL Server Profiler is an interface to create and manage traces and analyze and replay trace results. Events are saved in a trace file that can later be analyzed or used to replay a specific series of steps when trying to diagnose a problem.

Analyze query in Database Engine Tuning Advisor using the context menu

The Microsoft Database Engine Tuning Advisor (DTA) analyzes databases and makes recommendations that you use to optimize query performance. You use the Database Engine Tuning Advisor to select and create an optimal set of indexes, indexed views, or table partitions without having an expert understanding of the database structure or the internals of SQL Server. Using the DTA, you can perform the following tasks.

Design query in editor using the context menu

The Query and View Designer opens when you open the definition of a view, show the results for a query or view, or create or open a query.

Include actual execution plan using the context menu

Executes the query, returns the query results, and uses the execution plan for the query. The queries appear as a graphical query plan in the **Execution plan** window.

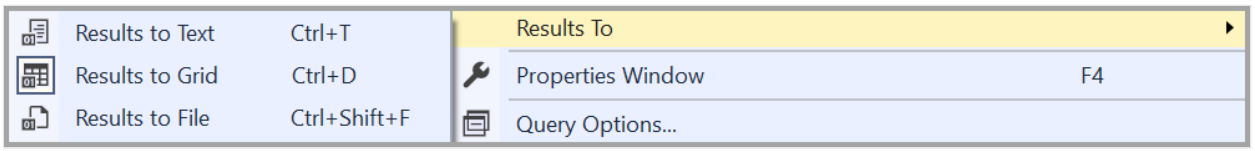
Include live query statistics using the context menu

Provides real-time insights into the query execution process as the controls flow from one query plan operator to another.

Include client statistics using the context menu

Includes a **Client Statistics** window that contains statistics about the query and about the network packets, and the elapsed time of the query.

Results using the context menu



You can select any of the *Result* options you want from the context menu.

- **Results to Text** - Returns the query results as text in the **Results** window.
- **Results to Grid** - Returns the query results as one or more grids in the **Results** window.
- **Results to File** - When the query executes, the **Save Results** dialog box opens. In **Save In**, select the folder in which you want to save the file. In **File name**, type the file's name, and then select **Save** to save the query results as a **Report** file with the .rpt extension. For advanced options, select the down-arrow on the **Save** button, and then select **Save with Encoding**.

Properties window using the context menu

The Properties window describes the state of an item in SQL Server Management Studio, such as a connection or a Showplan operator, and information about database objects such as tables, views, and designers.

You use the Properties window to view the properties of the current connection. Many properties are read-only in the Properties window but can be changed elsewhere in the Management Studio. For example, the Database property of a query is read-only in the Properties window but can be changed on the tool bar.

Query options using the context menu

Opens the **Query Options** dialog box. Use this dialog box to configure the default options for query execution and query results.

Transact-SQL F1 help

The Query Editor supports linking you to the reference article for a specific Transact-SQL statement when you select F1. To do so, highlight the name of a Transact-SQL statement

and then select F1. The help search engine then searches for an article that has an F1 help attribute that matches the string you highlighted.

If the help search engine doesn't find an article with an F1 help keyword that exactly matches the string you highlighted, then this article is displayed. In that case, there are two approaches to finding the help you're looking for:

- Copy and paste the editor string you highlighted into the search tab of SQL Server Books Online and do a search.
- Highlight only the part of the Transact-SQL statement likely to match an F1 help keyword applied to an article and select F1 again. The search engine requires an exact match between the string you highlighted and an F1 help keyword assigned to an article. If the string you highlighted contains elements unique to your environment, such as column or parameter names, the search engine doesn't get a match. Examples of the strings to highlight include:
  - The name of a Transact-SQL statement, such as `SELECT`, `CREATE DATABASE`, or `BEGIN TRANSACTION`.
  - The name of a built-in function, such as `SERVERPROPERTY`, or `@@VERSION`.
  - The name of a system stored procedure table, or views, such as `sys.data_spaces` or `sp_tableoption`.

Editor tasks

Expand table

Task Description	Article
Describes the various ways you can open the editors in SSMS.	Open an Editor (SQL Server Management Studio)
Configure options for the various editors, such as line numbering and IntelliSense options.	Configure Editors (SQL Server Management Studio)
How to manage the view mode, such as word wrap, splitting a window, or tabs.	Manage the Editor and View Mode
Set formatting options, such as hidden text or indentation.	Manage Code Formatting

Task Description	Article
Navigate through the text in an editor window by using features such as incremental search or go to.	Navigate Code and Text
Set color coding options for various classes of syntax, which makes it easier to read complex statements.	Color Coding in Query Editors
Drag text from one location in a script and drop it in a new location.	Drag and Drop Text
How to set bookmarks in order to more easily find important pieces of code.	Manage Bookmarks
How to print scripts or the results in a window or grid.	Print Code and Results
View and use the basic features in the MDX Query Editor.	Create Analysis Services Scripts
View and use the basic features in the DMX Query Editor.	Create a DMX Query
View and use the basic features in the XML/A Editor.	XML Editor (SQL Server Management Studio)
How to use the sqlcmd features in the Database Engine Query Editor.	Edit SQLCMD Scripts with Query Editor
How to use code snippets in the Database Engine Query Editor. Snippets are templates for commonly used statements or blocks, and can be customized or extended to include site-specific snippets.	Add Transact-SQL Snippets
How to use the Transact-SQL debugger to step through code and view debugging information such as the values in variables and parameters.	Transact-SQL debugger

Related content

- Customize Menus and Shortcut Keys
- SQL Server Management Studio keyboard shortcuts

Feedback

Was this page helpful? Yes No

Connect to Microsoft Azure Storage

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use the **Azure Storage Connection** dialog to specify a storage account and validate your connection to Azure.

Options

Specify the following information about your Azure account, and then select **Next** to continue.

1. **Storage Account** - Specify the storage account name.

   You can only connect to General-purpose Storage Accounts. Connecting to other types of storage accounts can result in an error similar to the following example:

   > The value for one of the HTTP headers is not in the correct format. (Microsoft.SqlServer.StorageClient).
   >
   > The remote server returned an error: (400) Bad Request. (System)

2. **Account Key** - Specify the account key for the specified storage account.
3. **Use secure endpoints (HTTPS)** - This option utilizes encrypted communication and secure identification of a network web server.
4. **Save account key** - This option saves your password in an encrypted file.

Feedback

Was this page helpful? Yes No

Connect to any SQL Server component from SQL Server Management Studio

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

SQL Server Management Studio provides functionality for managing every component of SQL Server. Use Management Studio to connect to:

- An instance of the SQL Server Database Engine.
- Analysis Services.
- Integration Services.
- Reporting Services.

Although Management Studio allows you to work with queries without first establishing a connection to a data source, most other tasks require a connection. Management Studio provides the **Connect to Server** dialog box to configure connection properties to SQL Server components. When Management Studio starts, it opens the **Connect to Server** dialog box and prompts you to connect to a server. The **Connect to Server** dialog box retains the connection settings from the last time it was used.

> **Note**
>
> This feature can be turned off so no connection is automatically initiated. For more information, see Database Engine Service startup options.

Save connections

You can save connections to specific servers in Registered Servers, or you can save connections in projects with Solution Explorer.

Save connections in Registered Servers

When you register a server, Management Studio saves the connection information in Registered Servers. To connect to a registered server, double-click the server name in Registered Servers. Object Explorer then opens a connection to the server.

Save connections in Solution Explorer

Solution Explorer allows you to store related queries, scripts, connections, and other associated information in a project. Each script project contains a node called **Connections**, where you can save one or more connections. To add a connection, right-click **Connections**, and then select **New Connection**. To access a saved connection, expand **Connections** and double-click the connection. Management Studio opens a query window associated with that connection. When saved, scripts retain their association to a specific connection.

Related content

- What is SQL Server Management Studio (SSMS)?
- Object Explorer

Feedback

Was this page helpful? Yes No

Connect to Server (Login page) - Database Engine

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use this tab to configure the connection properties when you connect to Microsoft SQL Server Database Engine. In most cases, you can connect by entering the computer name of the database server in the **Server name** box and then selecting **Connect**. If you're connecting to a named instance, use the computer name followed by a backslash and then the instance name. For example, `mycomputer\myinstance`. See Connect to the Database Engine for more examples.

> **Note**
>
> While Microsoft Entra ID is the new name for Azure Active Directory (Azure AD), to prevent disrupting existing environments, Azure AD still remains in some hardcoded elements such as UI fields, connection providers, error codes, and cmdlets. In this article, the two names are interchangeable.

Many factors can affect your ability to connect to SQL Server. For help, see the following resources:

- Lesson 1: Connecting to the Database Engine
- Troubleshoot Connecting to the SQL Server Database Engine
- Troubleshoot connectivity issues in SQL Server

> **Note**
>
> To connect with SQL Server Authentication, SQL Server must be configured in SQL Server and Windows Authentication mode. For more information about determining the authentication mode and changing the authentication mode, see Change server authentication mode.

Connect to a server

You can connect to any supported server from the login page by providing the server name and authentication details, as noted in the following sections.

Server type

When you register a server from Object Explorer, select the type of server to connect to: Database Engine, Analysis Services, Reporting Services, or Integration Services. The dialog box only shows the options for the selected server type. When you register a server from **Registered Servers**, the **Server type** box is read-only and matches the server type displayed in the **Registered Servers** component. To register a different type of server, select Database Engine, Analysis Services, Reporting Services, SQL Server Compact, or Integration Services from the **Registered Servers** toolbar before you start to register a new server.

Server name

Select the server instance you want to connect to. The most recent server instance you connected to, appears by default.

To connect to an instance of SQL Server Express, connect using the named pipes protocol specifying the pipe name, such as `np:\\.\pipe\3C3DF6B1-2262-47\tsql\query`. For more information, see the SQL Server Express documentation.

> **Note**
>
> Connections are persisted in the Most Recently Used (MRU) history. Select the **Server name** dropdown list to remove entries from the MRU, hover over the server name to remove, and then select the **Delete** key on your keyboard.

When you connect to Azure SQL Database, you can specify a database in the **Connect to Server** dialog box on the **Connection Properties** tab, which you can access by selecting **Options>>**.

By default, you connect to the `master` database. If you specify a user database when you connect to Azure SQL Database, you only see that database and its objects in Object Explorer. If you connect to `master`, you can see all databases. For more information, see the Microsoft Azure SQL Database Overview.

Authentication

The current version of SQL Server Management Studio (SSMS) offers eight authentication modes when you connect to a Database Engine. If your **Authentication** dialog box doesn't match the following list, download the most recent version of SQL Server Management Studio.

- **Windows Authentication**: Microsoft Windows Authentication mode allows a user to connect through a Windows user account.
- **SQL Server Authentication**: When you connect with a specified login name and password, SQL Server performs the authentication by checking to see if a matching SQL Server login account exists and if the specified password matches the one previously recorded. If the SQL Server login doesn't exist, authentication fails, and you receive an error message.
- **Microsoft Entra MFA**: This interactive workflow supports Microsoft Entra multifactor authentication (MFA). MFA helps safeguard access to data and applications while meeting user demand for a simple sign-in process. It delivers strong authentication with various easy verification options like phone calls, text messages, smart cards with pins, and mobile app notifications. When the user account is configured for MFA, the interactive authentication workflow requires more user interaction through pop-up dialog boxes and smart card use. If the user account doesn't require MFA, you can still use the other Microsoft Entra authentication options. For more information, see Using Microsoft Entra multifactor authentication.
- **Microsoft Entra Password**: This method for connecting to Azure SQL Database uses identities and their passwords in Microsoft Entra ID. It's useful when your Windows login credentials aren't in an Azure federated domain, or the initial or client domain is using Microsoft Entra authentication. For more information, see Use Microsoft Entra authentication.
- **Microsoft Entra Integrated**: This mechanism for connecting to Azure SQL Database uses Windows identities federated with Microsoft Entra ID. Use this method for connecting to SQL Database if you're logged in to Windows and using your credentials from a federated domain, or a managed domain configured for seamless single sign-on for pass-through and password hash authentication. For more information, see Use Microsoft Entra authentication and Microsoft Entra seamless single sign-on.
- **Microsoft Entra Service Principal**: A service principal is a Microsoft Entra identity that you can create for use with automated tools, jobs, and applications. With Service Principal authentication, you can connect to your SQL instance by using the client ID and secret of a service principal identity. In SSMS, enter the client ID in the

User name field and the secret in the **Password** field. For more information, see Microsoft Entra server principals and Microsoft Entra service principal with Azure SQL.

- **Microsoft Entra Managed Identity**: Managed identities are special forms of service principals. There are two types of managed identities: system-assigned and user-assigned. System-assigned managed identities are enabled directly on an Azure resource. User-assigned managed identities are a standalone resource you can assign to one or more Azure resources.

  To use a managed identity to connect to a SQL resource through GUI clients such as SSMS and ADS, the machine running the client application must have a Microsoft Entra client running with the identity's certificate stored in it. This requirement is most commonly achieved through an Azure VM because you can assign the identity to the machine through the VM's portal pane. For more information, see Managed identities in Microsoft Entra for Azure SQL.

- **Microsoft Entra Default**: This option performs authentication based on passwordless and noninteractive mechanisms, including managed identities, Visual Studio, Visual Studio Code, and the Azure CLI.

User name

The user name to connect. This setting is read-only when you select **Windows Authentication** or **Microsoft Entra Integrated** authentication. The setting is prefilled with the current login with your Windows user name.

If you connect with **Microsoft Entra Universal with MFA**, **Microsoft Entra Password**, **Microsoft Entra Service Principal**, or **Microsoft Entra Default**, enter the name of the Microsoft Entra identity you're connecting with.

User assigned identity

This option appears when you connect with **Microsoft Entra Managed Identity**. Completing this option properly depends on the type of the identity.

Login

Enter the login you connect with. This option is only available if you connect using **SQL Server Authentication**.

Password

Enter the password for the login. This option is only editable if you choose to connect using **SQL Server Authentication**, **Microsoft Entra Service Principal**, or **Microsoft Entra Password** authentication.

Remember password

Select this option for SQL Server to store the password entered. This option is displayed for the same authentication methods mentioned for the **Password** option.

Encryption

Select the level of encryption for the connection. The options for SSMS 20 are *Strict (SQL Server 2022 and Azure SQL)*, *Mandatory*, and *Optional*. When enabled, SQL Server uses TLS encryption for all the data sent between the client and server. For more information, see SQL Server and client encryption summary.

The default value is *Mandatory* in SSMS 20. *Strict (SQL Server 2022 and Azure SQL)* encryption should be used for Azure SQL Database and Azure SQL Managed Instance. *Strict (SQL Server 2022 and Azure SQL)* encryption can be used for SQL Server when the instance has **Force Strict Encryption** enabled.

Mandatory encryption can be used for SQL Server when the instance has **Force Encryption** enabled. It can also be used when no encryption is configured for the instance, if **Trust server certificate** is enabled. While this method is less secure than installing a trusted certificate, it does support an encrypted connection.

The **Encryption** property appears on the Login page for SSMS 20.x and later versions.

Trust server certificate

When enabled, with *Optional* or *Mandatory* encryption selected, or if the server is configured to force encryption, SQL Server doesn't validate the server certificate on the client machine when encryption is enabled for the network communication between client and server.

The **Trust server certificate** property appears on the Login page for SSMS 20.x and later versions.

Host name in the certificate

The value provided in this option is used to specify a different, but expected, CN or SAN in the server certificate for the server to which SSMS is connecting. This option can be left blank, so that certificate validation ensures that the Common Name (CN) or Subject Alternate Name (SAN) in the certificate matches the server name to which you're connecting. This parameter can be populated when the server name doesn't match the CN or SAN, for example, when using DNS aliases. For more information, see Encryption and certificate validation in Microsoft.Data.SqlClient. |

The Encryption property appears on the Login page for SSMS 20.x and later versions.

Connect

Select to connect to the server.

Options

Select to collapse the connection dialog or expand the dialog to display the **Connection Properties**, **Always Encrypted**, and **Additional Connection Parameters** tabs.

Related content

- Connect to server (Connection Properties page) - Database Engine
- Connect to Server (Always Encrypted page) - Database Engine
- Connect to Server (Additional Connection Parameters page) - Database Engine

---

Feedback

Was this page helpful? Yes No

Connect to server (Connection Properties page) - Database Engine

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use this tab to view or specify options when you connect to an instance of Microsoft SQL Server Database Engine or when you register a Database Engine in **Registered Servers**. **Connect** and **Options>>** only appear in this dialog box when you connect to an instance of the Database Engine. **Test** and **Save** only appear in this dialog box when registering Database Engine. Access this tab by selecting **Options>>** on the **Login** tab.

> **Note**
>
> While Microsoft Entra ID is the new name for Azure Active Directory (Azure AD), to prevent disrupting existing environments, Azure AD still remains in some hardcoded elements such as UI fields, connection providers, error codes, and cmdlets. In this article, the two names are interchangeable.

Connect to database

Select a database to connect to from the list. If you select **<default>**, you connect to the default database for the server. If you select **<Browse server>**, you can browse the server for the database to which to connect.

When you connect to an instance of the SQL Server Database Engine through Azure SQL Database, you must use SQL Server Authentication and specify a database in the **Connect to Server** dialog box on the **Connection Properties** tab. Ensure that you select the **Encrypt connection** checkbox.

By default, SQL Server connects to `master`. When you connect to SQL Database, if you specify a user database, you only see that database and its objects in Object Explorer. If you connect to `master`, you can see all databases. For more information, see What is Azure SQL Database?.

Network protocol

Select a protocol from the list. The available client protocols are configured using the Client Network Configuration in Computer Management.

Network packet size

Enter the size of the network packets to be sent. The default is 4,096 bytes.

Connection time-out

Enter the number of seconds to wait for a connection to be established before timing out. The default value is 30 seconds.

Execution time-out

Enter the time in seconds to wait before the execution of a query is completed on the server. The default value is zero seconds, which indicates there's no time-out.

Encrypt connection

Forces encryption of the connection. When enabled, SQL Server uses TLS encryption for all the data sent between the client and server. For more information, see SQL Server and client encryption summary.

The **Encrypt connection** property appears on the Connection Properties for SSMS 19.x and earlier versions.

Trust server certificate

When enabled, with **Encrypt connection** also enabled or if the server is configured to force encryption, SQL Server doesn't validate the server certificate on the client machine when you enable encryption on the network connection between client and server.

The **Trust server certificate** property appears on the Connection Properties page for SSMS 19.x and earlier versions.

Use custom color

Select to specify the background color for the status bar in a Database Engine Query Editor window. To specify the color, choose **Select**. In the **Color** dialog box, select a predefined color from the **Basic Colors** grid or select **Define Custom Colors** to define and use a custom color.

- When you specify a color for a server entry in the **Object Explorer** pane, that color is used when you open a Query Editor window. To open a Query Editor window, right-click the server entry and select **New Query**. Alternatively, when the **Object Explorer** pane is active and focused on this server, you can select **New Query** on the toolbar.
- When you specify a color for a server entry in the **Registered Servers** pane, that color is used when you open a Query Editor window. To open a Query Editor window, right-click the server entry and select **New Query**. Alternatively, when the **Registered Server** pane is active and focused on this server, you can select **New Query** on the toolbar.
- On the **File** menu, when you select **New** > **Database Engine Query**, the color that you specify in the **Connect to Server** dialog box applies to that Query Editor window.

Reset All

Replace all manually entered connection property values with their defaults.

Connect

Attempt a connection using the listed values.

Options

Select to hide the **Connection Properties** and **Additional Connection Parameters** tabs and view the **Login** tab.

Test

When you register Database Engine in **Registered Servers**, select to test the connection.

Save

Saves the settings in **Registered Servers**.

Related content

- Quickstart: Connect and query a SQL Server instance using SQL Server Management Studio (SSMS)

- Quickstart: Connect and query a SQL Server instance on an Azure Virtual Machine using SQL Server Management Studio (SSMS)
- Quickstart: Connect and query an Azure SQL Database or an Azure SQL Managed Instance using SQL Server Management Studio (SSMS)
- Quickstart: Connect and query a dedicated SQL pool (formerly SQL DW) in Azure Synapse Analytics with SQL Server Management Studio (SSMS)
- Quickstart: Use the Azure portal query editor to query Azure SQL Database

---

Feedback

Was this page helpful? Yes No

Connect to Server (Always Encrypted page) - Database Engine

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance

Use this tab to view or specify Always Encrypted column encryption options when you connect to Azure SQL Database, Azure SQL Managed Instance, or SQL Server. Use this tab to view or specify Always Encrypted secure enclave options when you connect to Azure SQL Database or SQL Server. Access this tab by selecting **Options >>** on the **Login** tab.

Always Encrypted tab

The **Always Encrypted** tab has two checkboxes.

Enable Always Encrypted (column encryption)

By default, Always Encrypted is disabled for a database connection. Enabling Always Encrypted for a database connection instructs the .NET Framework Data Provider for SQL Server, used by SQL Server Management Studio (SSMS), to attempt to transparently:

- Decrypt any values that are retrieved from encrypted columns and returned in query results.
- Encrypt the values of the parameterized Transact-SQL variables that target encrypted database columns.

SSMS uses the .NET Framework Data Provider for SQL Server. If you don't enable Always Encrypted for a connection, the .NET Framework Data Provider doesn't try to encrypt query parameters or decrypt results. For more information, see Query columns using Always Encrypted with SQL Server Management Studio.

Enable secure enclaves

Enable this option when you connect to a database configured to use Always Encrypted with secure enclaves. For more information, see Getting started using Always Encrypted with secure enclaves.

Enclave attestation

The **Enclave attestation** section has two options.

Protocol

Select the attestation protocol to use when you connect to a database configured to use Always Encrypted with secure enclaves. The protocol value determines:

- If the client app uses attestation, and, if so,
- It specifies the type of the attestation service it uses.

Available options are **None**, **Host Guardian Service**, and **Microsoft Azure Attestation**. For more information, see Always Encrypted with secure enclaves.

URL

Enter the configured URL for the selected attestation protocol. The URL is specific to the type of enclave and SQL in use.

- Azure SQL Database with SGX enclaves
- SQL Server

Attestation isn't supported for Azure SQL Database with VBS enclaves. The URL option is disabled if **None** is selected for **Protocol**.

Related content

- Tutorial: Getting started using Always Encrypted with secure enclaves in SQL Server with attestation using HGS
- Tutorial: Getting started using Always Encrypted with Intel SGX enclaves in Azure SQL Database
- Tutorial: Getting started using Always Encrypted with VBS enclaves in Azure SQL Database

---

Feedback

Was this page helpful? Yes No

Connect to Server (Additional Connection Parameters page) - Database Engine

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use this tab to enter advanced parameters when connecting to an instance of Microsoft SQL Server Database Engine or registering Database Engine in **Registered Servers**. **Connect** and **Options** only appear in this dialog box when connecting to an instance of the Database Engine. **Test** and **Save** only appear in this dialog box when registering Database Engine. Access this tab by selecting **Options** >> on the login tab.

> **Note**
>
> While Microsoft Entra ID is the new name for Azure Active Directory (Azure AD), to prevent disrupting existing environments, Azure AD still remains in some hardcoded elements such as UI fields, connection providers, error codes, and cmdlets. In this article, the two names are interchangeable.

Additional parameters

The Additional Connection Parameters dialog box allows you to enter connection properties that don't have an equivalent option exposed in any other tab, usually to support an advanced scenario. The parameters entered in the Additional Connection Parameters dialog box are added to any options selected or altered in the **Login** and **Connection Properties** dialogs.

> **Important**
>
> Any parameters entered in this dialog box override selections made on the **Login** and **Connection Properties** dialogs. Parameters are transmitted as clear text. Don't include sensitive information in the dialog.

Add a parameter

Parameters are entered with the property name and followed by the value for the property. For scenarios that include Availability Groups, to connect to a secondary replica that supports read-intent so you can execute read-only queries, enter `ApplicationIntent=ReadOnly`. If you connect to an Availability Group with replicas in multiple subnets and specify the Listener in the **Server name** dialog on the **Login** page, enter `MultiSubnetFailover=True` for faster detection of, and connection to, the (currently) active server.

When entering multiple parameters, separate them with a semicolon. For example, `ApplicationIntent=ReadOnly; InitialCatalog=WideWorldImporters`.

For a complete list of parameters, see Properties.

> **⊗ Caution**
>
> Additional parameters aren't typically required and exist to support advanced scenarios. If you determine that a connection isn't working as expected, check the **Additional Connection Parameters** dialog to see if any unexpected options exist. It isn't uncommon that users enter options for testing purposes and then forget to remove them.

Related content

- Connect to Server (Login page) - Database Engine
- Connect to Server (Connection Properties page) - Database Engine

---

Feedback

Was this page helpful? 👍 Yes 👎 No

Connect to SQL Server Analysis Services

Article • 12/09/2022

Applies to: SQL Server Analysis Services Azure Analysis Services Fabric/Power BI Premium

This article describes connecting to an instance of SQL Server Analysis Services (SSAS) using common tools and applications. This article also explains how to connect under different user identities for testing purposes.

To learn about connecting to **Azure Analysis Services**, see Connecting to server resources

To learn about connecting to **Power BI Premium workspaces**, see Connecting to a Premium workspace

Firewall and permissions

Successful connections to SSAS depend on a valid port configuration and appropriate user permissions. Click the following links to learn more about each requirement.

- Configure the Windows Firewall to Allow Analysis Services Access
- Authorizing access to objects and operations (Analysis Services)

> **Note**
>
> The client libraries required by client applications cannot connect to Analysis Services through proxy servers that require a username and password.

Connect using SQL Server Management Studio (SSMS)

Connect to Analysis Services in SSMS to manage server instances and databases interactively. You can also run XMLA or MDX queries to perform administrative tasks or retrieve data. In contrast with other tools and applications that only load databases when a query is sent, SSMS loads all databases when you connect to the server, assuming you have permission to view the database. This means that if you have numerous tabular databases on the server, all are loaded into system memory when you connect using SSMS.

You can test permissions by running SSMS under a specific user identity and then connect to Analysis Services as that user.

Hold-down the Shift key and right-click the **SQL Server Management Studio** shortcut to access the **Run as different user** option.

1. Start SQL Server Management Studio. In the **Connect to Server** dialog box, select the SQL Server Analysis Services server type.

2. In the Login tab, enter the server name by typing the name of the computer on which the server is running. You can specify the server using its network name or a fully-qualified domain name.

   For a named instance, the server name must be specified in this format: servername\instance name. An example of this naming convention might be ADV-SRV062\Finance for a server that has a network name of ADV-SRV062, where Analysis Services was installed as a named instance entitled Finance.

   For servers deployed in a failover cluster, connect using the network name of the SSAS cluster. This name is specified during SQL Server setup, as **SQL Server Network Name**. Note that if you installed SSAS as a named instance onto a Windows Server Failover Cluster (WSFC), you never add the instance name on the connection. This practice is unique to SSAS; in contrast, a named instance of a clustered relational database engine does include the instance name. For example, if you installed both SSAS and the database engine as named instance (Contoso-Accounting) with a SQL Server Network Name of SQL-CLU, you would connect to SSAS using "SQL-CLU" and to the database engine as "SQL-CLU\Contoso-Accounting". See How to Cluster SQL Server Analysis Services for more information and examples.

   For servers deployed in a network load balanced cluster, connect using the virtual server name of the NLB.

3. Authentication is always Windows authentication, and the user identity is always the Windows user who is connecting via Management Studio.

   In order for the connection to succeed, you must have permission to access the server or a database on the server. Most tasks that you want to perform in Management Studio require administrative permissions. Ensure that the account you are connecting with is a member of the Server Administrator role. For more information, see Grant server admin rights to an Analysis Services instance.

4. Click **Connection Properties** to specify a particular database, set timeout values, or encryption options. Optional connection information includes connection properties used for the current connection only.

5. Click **Additional Connection Parameters** tab to set connection properties not available in the Connect to Server dialog box. For example, you might type `Roles=Reader` in the text

box.

Connecting through a role with less permission lets you test database behaviors when that role is in effect.

```
Provider=MSOLAP; Data Source=SERVERNAME; Initial Catalog=AdventureWorks2012;
Roles=READER
```

Connect using Excel

Microsoft Excel is often used for analyzing business data. As part of an Excel installation, Office installs the Analysis Services OLE DB provider (MSOLAP DLL), ADOMD.NET, and other data providers so that you can more readily use the data on your network servers. If you are using a newer version of SQL Server Analysis Services with an older version of Excel, you most likely need to install newer client libraries on each workstation that connects to SQL Server Analysis Services. To learn more, see Client libraries.

When you set up a connection to an Analysis Services cube or tabular model database, Excel saves the connection information in .odc file for future use. The connection is made in security context of the current Windows user. The user account must have read permissions on the database in order for the connection to succeed.

When using SQL Server Analysis Services data in an Excel workbook, connections are held for the duration of a query request. This is why you are likely to see lots of connections for each session, held for very short periods of time, when monitoring a query workload from Excel.

You can test permissions by starting Excel under a specific user identity.

Hold-down the Shift key and right-click the **Excel** shortcut to access the **Run as different user** option.

1. On the Data tab in Excel, click **From Other Sources**, and then click **From Analysis Services**. Enter the server name, and then select a cube or perspective to query.

   For servers deployed in a load-balanced cluster, use the virtual server name assigned to the cluster.

2. When setting up a connection in Excel, on the last page of the Data Connection Wizard, you can specify authentication settings for Excel Services. These settings are used to set properties on the workbook should you upload it to a SharePoint server that has Excel

Services. The settings are used in data refresh operations. Options include **Windows Authentication**, **Secure Store Service** (SSS), and **None**.

Avoid using **None**. Analysis Services does not let you specify a user name and password on the connection string unless you are connecting to a server that has been configured for HTTP access. Similarly, do not use SSS unless you already know that the SSS target application ID is mapped to a set of Windows user credentials that have user access to the Analysis Services databases. For most scenarios, using the default option of Windows authentication is the best choice for an Analysis Services connection from Excel.

For more information, see Connect to or import data from SQL Server Analysis Services.

Connect using Visual Studio

Visual Studio with Analysis Services projects is used for building BI solutions. When building reports or packages, you might need to specify a connection to Analysis Services.

The following links explain how to connect to Analysis Services from a Report Server project or an Integration Services project:

- Analysis Services Connection Type for MDX (SSRS)
- Analysis Services Connection Manager

> **Note**
>
> When using Visual Studio to work on an existing Analysis Services project, remember that you can connect offline using a local or version controlled project, or connect in online mode to update Analysis Services objects while the database is running. For more information, see **Connect in Online Mode to an Analysis Services Database**. More commonly, connections from SQL Server Data Tools are in project mode, where changes are deployed to the database only when you explicitly deploy the project.

Test connections

Use SQL Server Profiler to monitor connections to Analysis Services. The Audit Login and Audit Logout events provide evidence of a connection. The identity column indicates the security context under which the connection is made.

1. Start **SQL Server Profiler** on the Analysis Services instance and then start a new trace.

2. In Events Selection, verify that **Audit Login** and **Audit Logout** are checked in the Security Audit section.

3. Connect to Analysis Services via an application service (such as SharePoint or Reporting Services) from a remote client computer. The Audit Login event will show the identity of the user connecting to Analysis Services.

Connection errors are often traced to an incomplete or invalid server configuration. Always check server configuration first:

- Ping the server from a remote computer to ensure it allows remote connections.
- **Firewall rules on the server allow inbound connections from clients in the same domain**

  With the exception of Power Pivot for SharePoint, all connections to a remote server require that you have configured the firewall to allow access to the port that Analysis Services is listening on. If you are getting connection errors, verify that the port is accessible and that user permissions are granted to the appropriate databases.

  To test, use Excel or SSMS to on a remote computer, specifying the IP address and port used by the Analysis Services instance. If you can connection, the firewall rules are valid for the instance and the instance allows remote connections.

  Also, when using TCP/IP for the connection protocol, remember that Analysis Services requires client connections originate from the same domain or a trusted domain. If connections flow across security boundaries, you will most likely need to configure HTTP access. For more information, see Configure HTTP Access to Analysis Services on Internet Information Services (IIS) 8.0.

- Can you connect using some tools but not others? The problem might be the wrong version of a client library. You can get client libraries from the SQL Server Feature Pack download page.

Resources that can help you resolve connection failures include the following:

Resolving Common Connectivity Issues in SQL Server 2005 Analysis Services Connectivity Scenarios. This document is a few years old, but the information and methodologies still apply.

See Also

Authentication methodologies supported by Analysis Services
Impersonation
Create a Data Source (SSAS Multidimensional)

Connect to a report server in Management Studio

Article • 12/17/2024

SQL Server Management Studio provides Object Explorer, so you can connect to any server in the SQL Server family and graphically browse its contents. Object Explorer allows you to open connections to multiple server instances in the same workspace as long as the servers are registered in the same server group. For Reporting Services, you can use Object Explorer to do the following tasks:

- Enable report server features.
- Set server defaults and configure role definitions.
- Manage jobs that are running.
- Manage job schedules.

Prerequisites

- SQL Server Management Studio.
- A configured report server in native or Sharepoint mode. For instructions on how to configure your report server, see Configure a report server (Reporting Services native mode).
- Sufficient permissions to manage your Report Server. For more information about permissions, see Connection syntax and permissions.
- The server must be registered before you can connect to a report server instance in Management Studio. For instructions to register a report server, see Register a report server.

Connect to a native mode report server

1. Open **Object Explorer** by selecting it from the **View** menu, if it isn't already open.
2. Select **Connect** to view the list of server types, and then choose the **Reporting Services...** option.
3. In the **Connect to Server** dialog box, enter the name of the report server instance into the **Server name** field. Report server instance names are based on SQL Server instance names. By default, the instance name of a local report server instance is the computer name. If

you install the report server as a named instance, use this syntax to specify the server: `<servername>[\<instancename>]`.

4. From the **Authentication** drop-down menu, select the authentication type your server uses. If you use **Windows Authentication**, you connect by using your credentials. If you select **Basic Authentication** or **Forms Authentication**, enter the account and password.

5. Select **Connect**. The report server appears in **Object Explorer**.

6. Right-click the server node to set system properties and server defaults. For more information, see Set report server properties (Management Studio).

Connect to a SharePoint integrated mode report server

1. Open **Object Explorer** by selecting it from the **View** menu, if it isn't already open.

2. Select **Connect** to view the list of server types, and then choose the **Reporting Services...** option.

3. In the **Connect to Server** dialog box, enter a URL to a SharePoint site into the **Server name** field. The following example illustrates the syntax: `https://<web server>/sites/<site>`.

4. From the **Authentication** drop-down menu, select the authentication type your server uses. If you use **Windows Authentication**, you must connect by using your credentials. If you select **Basic Authentication** or **Forms Authentication**, enter the account and password.

5. Select **Connect**. The report server appears in **Object Explorer**.

6. Right-click the server node to set system properties and server defaults. For more information, see Set report server properties (Management Studio).

Register a report server

1. If you can't connect to a report server, you either don't have permission to access it, or the server isn't registered. To register the server, select **View** in the menu > **Registered Servers**.

2. Select the **Reporting Services** icon.

3. Expand **Reporting Services** and right-click **Local Server Groups**. Select the **New Server Registration...** option. The **New Server Registration** dialog box appears.

4. For **Server name**, enter a value. You specify the value depending on the server mode:

    - For a native mode report server, enter the name of the report server instance. Report server instance names are based on SQL Server instance names. By default, the instance name of a local report server instance is the computer name. If you installed the report server as a named instance, use this syntax to specify the server: `<servername>[\<instancename>]`.

    - For a report server that runs in SharePoint integrated mode, you connect to the SharePoint site that the report server is connected to. Connect to the SharePoint site so that you can view the permission levels. The permissions control access to report server content and operations. You can specify any site in the site collection. The following example shows the syntax: `https://mysharepointsite`.

5. For **Authentication**, select the authentication mode that the report server uses from the drop-down menu.

    - If you use default security, choose **Windows Authentication**.
    - If you installed and deployed a custom security extension, choose **Forms Authentication**.
    - If you configured the report server to use basic authentication, choose **Basic Authentication**.
    - If you configured the report server for SharePoint integrated mode, choose **Windows Authentication**.

6. Select **Test** to verify the connection.

7. When prompted, select **OK**. Then select **Save**.

Connection syntax and permissions

When you specify Reporting Services as the **Server Type** in the **Connect to Server** dialog box, you can specify either a report server name or an endpoint to the Web service. The following table summarizes the connection syntax, steps, and permissions required to perform specific tasks:

Expand table

Connect to	Tasks	Permissions
Native mode report server, connected as the default, or named instance: `<server name>\<_instance>` The connection to the report server is made through the Report Server WMI (Windows Management Instrumentation) provider.	View and set server properties and defaults. View and cancel jobs. Create and manage shared schedules. Create, modify, or delete role definitions.	Assigned to the System Administrator role.
Native mode report server, connected as the default, or named instance, through the endpoint to the Report Server Web service: `https://<servername>/reportserver` Specifying a URL to the report server provides an alternate way to connect to the report server.	View and set server properties and defaults. View and cancel jobs. Create and manage shared schedules. Create, modify, or delete role definitions.	Assigned to the System Administrator role.
SharePoint integrated mode report server, connected through the SharePoint site: `https://<webserver>/<SharePointSite>`	View and set server properties and defaults. View and cancel jobs. Create and manage shared schedules defined for the site to which you're connected.	Full Control level of permission on the SharePoint site to which you're connected.

Connect to	Tasks	Permissions
	View the permission levels defined for the site to which you're connected.	
SharePoint integrated mode report server, connected through the name of the report server instance: `<server name>\<_instance>`	View and set server properties and defaults. View and cancel jobs.	Full Control level of permission on the SharePoint site that is integrated with the report server. Notice that when you connect to the report server rather than the SharePoint site, the number of tasks that you can do is reduced. The report server can only return application data that is stored or managed in the report server database. It can't return data stored in the SharePoint configuration and content databases.

Related content

- Configure a report server database connection (Report Server Configuration Manager)
- Reporting Services in SQL Server Management Studio (SSRS)

SQL Server Browser service

Article • 02/13/2025

Applies to: SQL Server - Windows only

The SQL Server Browser program runs as a Windows service. SQL Server Browser listens for incoming requests for SQL Server resources and provides information about SQL Server instances installed on the computer. SQL Server Browser contributes to the following actions:

- Browsing a list of available servers.
- Connecting to the correct server instance.
- Connecting to dedicated administrator connection (DAC) endpoints.

For each instance of the Database Engine and SQL Server Analysis Services (SSAS), the SQL Server Browser service (sqlbrowser) provides the instance name and the version number. SQL Server Browser is installed with SQL Server.

SQL Server Browser can be configured during setup or using SQL Server Configuration Manager. By default, the SQL Server Browser service starts automatically:

- When upgrading an installation.
- When installing on a cluster.
- When installing a named instance of the Database Engine, including all instances of SQL Server Express.
- When installing a named instance of SSAS.

Background

Before SQL Server 2000 (8.x), only one instance of SQL Server could be installed on a computer. SQL Server listened for incoming requests on port 1433, assigned to SQL Server by the official Internet Assigned Numbers Authority (IANA). Only one instance of SQL Server can use a port, so when SQL Server 2000 (8.x) introduced support for multiple instances of SQL Server, SQL Server Resolution Protocol (SSRP) was developed to listen on User Datagram Protocol (UDP) port 1434. This listener service responds to client requests with the names of the installed instances and the ports or named pipes used by the instance.

To resolve limitations of the SSRP system, SQL Server 2005 (9.x) introduced the SQL Server Browser service as a replacement for SSRP.

How SQL Server Browser works

When an instance of SQL Server starts and the TCP/IP protocol is enabled for SQL Server, the server is assigned a TCP/IP port. SQL Server listens on a specific named pipe if the named pipes protocol is enabled. This port or named pipe is used by that specific instance to exchange data with client applications. TCP/IP port 1433 and pipe `\sql\query` are assigned to the default instance during installation. The server administrator can change the port or named pipe using SQL Server Configuration Manager.

Because only one instance of SQL Server can use a port or pipe, different port numbers and pipe names are assigned for named instances, including SQL Server Express. When enabled, named instances and SQL Server Express are configured to use dynamic ports by default. That is, an available port is assigned when SQL Server starts.

If you want, a specific port can be assigned to an instance of SQL Server. Clients can specify a specific port when connecting to SQL Server. However, if the port is dynamically assigned, the port number can change anytime the SQL Server is restarted, so the correct port number is unknown to the client.

Upon startup, the SQL Server Browser starts and claims UDP port 1434. SQL Server Browser reads the registry, identifies all instances of SQL Server on the computer, and notes the ports and named pipes they use. When a server has two or more network cards, SQL Server Browser returns the first enabled port it encounters for SQL Server. SQL Server Browser supports ipv6 and ipv4.

When SQL Server clients request SQL Server resources, the client network library sends a UDP message to the server using port 1434. SQL Server Browser responds with the TCP/IP port or named pipe of the requested instance. The network library on the client application then completes the connection by sending a request to the server using the port or named pipe of the desired instance.

For more information on starting and stopping the SQL Server Browser service, see Start, stop, pause, resume, and restart SQL Server services.

Use SQL Server Browser

If the SQL Server Browser service isn't running, you can still connect to the SQL Server if you provide the correct port number or named pipe. For instance, you can connect to the default instance of SQL Server with TCP/IP if it's running on port 1433.

However, if the SQL Server Browser service isn't running, the following connections don't work:

- Any component that tries to connect to a named instance without fully specifying all the parameters (such as the TCP/IP port or named pipe).

- Any component that generates or passes `<server>\<instance>` information that other components could later use to reconnect.
- Connecting to a named instance without providing the port number or pipe.
- Diagnostic connection for database administrators to a named instance or the default instance if not using TCP/IP port 1433.
- The Online Analytical Processing (OLAP) redirector service.
- Enumerating servers in SQL Server Management Studio.

Suppose you're using SQL Server in a client-server scenario (for example, when your application is accessing SQL Server across a network). If you stop or disable the SQL Server Browser service, you must assign a specific port number to each instance and configure your client application code to use that port number. This approach has the following problems:

- You must update and maintain client application code to ensure it's connecting to the proper port.
- The port you choose for each instance could be used by another service or application on the server, causing the instance of SQL Server to be unavailable.

Clusters and SQL Server Browser

SQL Server Browser isn't a clustered resource and doesn't support failover from one cluster node to the other. Therefore, if there's a cluster, SQL Server Browser should be installed and turned on for each cluster node. On clusters, the SQL Server Browser listens on `IP_ANY`.

When the SQL Server Browser listens on `IP_ANY` and you enable listening on specific IP addresses, you must configure the same TCP/IP port on each IP address, because SQL Server Browser returns the first IP address and port pair it encounters.

Install, uninstall, and run from the command line

By default, the SQL Server Browser program is installed at `<drive>:\Program Files (x86)\Microsoft SQL Server\<nn>\Shared\sqlbrowser.exe`.

The SQL Server Browser service is uninstalled when the last instance of SQL Server is removed.

SQL Server Browser can be started from the command prompt for troubleshooting by using the `-c` switch:

```
```

Windows Command Prompt

```
<drive>\<path>\sqlbrowser.exe -c
```

Security

The SQL Server Browser Service is crucial in facilitating network communication with SQL Server instances.

Security measures for SQL Server Browser Service include:

- Configuring firewalls to allow its traffic.
- Restricting access to trusted IP addresses.
- Regularly applying updates to patch vulnerabilities.
- Additionally, you must implement strong authentication and authorization policies to prevent unauthorized access and maintain the integrity of your SQL Server environment.

Account privileges

SQL Server Browser listens on a UDP port and accepts unauthenticated requests using SQL Server Resolution Protocol (SSRP). SQL Server Browser should be run in the security context of a low-privileged user to minimize exposure to a malicious attack. The sign in account can be changed by using the SQL Server Configuration Manager.

The minimum user rights for SQL Server Browser are:

- Deny access to this computer from the network.
- Deny sign in locally.
- Deny sign in as a batch job.
- Deny Log On Through Terminal Services.
- Sign in as a service.
- Read and write the SQL Server registry keys related to network communication (ports and pipes).

Default account

Setup configures SQL Server Browser to use the account selected for services during setup. Other possible accounts include:

- Any **domain\local** account.

- The **Local Service** account.
- The **Local System** account (not recommended as it has unnecessary privileges).

Hide SQL Server

Hidden instances are instances of SQL Server that support only shared memory connections. For SQL Server, set the `HideInstance` flag to indicate that SQL Server Browser shouldn't respond with information about this server instance.

Use a firewall

To communicate with the SQL Server Browser service on a server behind a firewall, open UDP port 1434 and the TCP/IP port used by SQL Server (for example, 1433). For information about working with a firewall, see Configure the Windows Firewall to Allow SQL Server Access.

Related content

- Network Protocols and Network Libraries
- Connecting Using IPv6
- Enable or disable a server network protocol

Using Connection String Keywords with SQL Server Native Client

Article • 01/10/2024

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

> **Important**
>
> SQL Server Native Client (SNAC) isn't shipped with:
>
> - SQL Server 2022 (16.x) and later versions
> - SQL Server Management Studio 19 and later versions
>
> The SQL Server Native Client (SQLNCLI or SQLNCLI11) and the legacy Microsoft OLE DB Provider for SQL Server (SQLOLEDB) aren't recommended for new application development.
>
> For new projects, use one of the following drivers:
>
> - Microsoft ODBC Driver for SQL Server
> - Microsoft OLE DB Driver for SQL Server
>
> For SQLNCLI that ships as a component of SQL Server Database Engine (versions 2012 through 2019), see this Support Lifecycle exception.

Some SQL Server Native Client APIs use connection strings to specify connection attributes. Connection strings are lists of keyword and associated values; each keyword identifies a particular connection attribute.

For information, see Using Connection String Keywords with OLE DB Driver for SQL Server.

> **Note**
>
> SQL Server Native Client allows ambiguity in connection strings to maintain backward compatibility (for example, some keywords may be specified more than once, and conflicting keywords may be allowed with resolution based on position or precedence). It is good practice when modifying applications to use SQL Server Native Client to eliminate any dependency on connection string ambiguity.

The following sections describe the keywords that can be used with the SQL Server Native Client OLE DB provider, the SQL Server Native Client ODBC driver, and ActiveX Data Objects (ADO) when using SQL Server Native Client as the data provider.

ODBC Driver connection string keywords

ODBC applications use connection strings as a parameters to the SQLDriverConnect and SQLBrowseConnect functions.

Connection strings used by ODBC have the following syntax:

```
connection-string ::= empty-string[;] | attribute[;] | attribute; connection-string
```

```
empty-string ::=
```

```
attribute ::= attribute-keyword=[{]attribute-value[}]
```

```
attribute-value ::= character-string
```

```
attribute-keyword ::= identifier
```

Attribute values can optionally be enclosed in braces, and it is good practice to do so. This avoids problems when attribute values contain non-alphanumeric characters. The first closing brace in the value is assumed to terminate the value, so values cannot contain closing brace characters.

The following table describes the keywords that may be used with an ODBC connection string.

Expand table

Keyword	Description
Addr	Synonym for "Address".
Address	The network address of the server running an instance of SQL Server. **Address** is usually the network name of the server, but can be other names such as a pipe, an IP address, or a TCP/IP port and socket address. If you specify an IP address, make sure that the TCP/IP or named pipes protocols are enabled in SQL Server Configuration Manager. The value of **Address** takes precedence over the value passed to **Server** in ODBC connection strings when using SQL Server Native Client. Also note that `Address=;` will connect to the server specified in the **Server** keyword, whereas `Address= ;, Address=.;`, `Address=localhost;`, and `Address=(local);` all cause a connection to the local server.

Keyword	Description
	The complete syntax for the **Address** keyword is as follows: [*protocol*:]*Address*[,*port* |*pipe**pipename*] *protocol* can be **tcp** (TCP/IP), **lpc** (shared memory), or **np** (named pipes). For more information about protocols, see Configure Client Protocols. If neither *protocol* nor the **Network** keyword is specified, SQL Server Native Client will use the protocol order specified in SQL Server Configuration Manager. *port* is the port to connect to, on the specified server. By default, SQL Server uses port 1433.
AnsiNPW	When "yes", the driver uses ANSI-defined behaviors for handling NULL comparisons, character data padding, warnings, and NULL concatenation. When "no", ANSI defined behaviors are not exposed. For more information about ANSI NPW behaviors, see Effects of ISO Options.
APP	Name of the application calling SQLDriverConnect (optional). If specified, this value is stored in the **master.dbo.sysprocesses** column **program_name** and is returned by sp_who and the APP_NAME functions.
ApplicationIntent	Declares the application workload type when connecting to a server. Possible values are **ReadOnly** and **ReadWrite**. The default is **ReadWrite**. For example: `ApplicationIntent=ReadOnly` For more information about SQL Server Native Client's support for Always On availability groups, see SQL Server Native Client Support for High Availability, Disaster Recovery.
AttachDBFileName	Name of the primary file of an attachable database. Include the full path and escape any \ characters if using a C character string variable: `AttachDBFileName=c:\\MyFolder\\MyDB.mdf` This database is attached and becomes the default database for the connection. To use **AttachDBFileName** you must also specify the database name in either the SQLDriverConnect DATABASE parameter or the SQL_COPT_CURRENT_CATALOG connection attribute. If the database was previously attached, SQL Server does not reattach it; it uses the attached database as the default for the connection.
AutoTranslate	When "yes", ANSI character strings sent between the client and server are translated by converting through Unicode to minimize problems in matching extended characters between the code pages on the client and the server. Client SQL_C_CHAR data sent to a SQL Server **char**, **varchar**, or **text** variable, parameter, or column is converted from character to Unicode using the client

Keyword	Description
	ANSI code page (ACP), then converted from Unicode to character using the ACP of the server. SQL Server **char**, **varchar**, or **text** data sent to a client SQL_C_CHAR variable is converted from character to Unicode using the server ACP, then converted from Unicode to character using the client ACP. These conversions are performed on the client by the SQL Server Native Client ODBC driver. This requires that the same ANSI code page (ACP) used on the server be available on the client. These settings have no effect on the conversions that occur for these transfers: * Unicode SQL_C_WCHAR client data sent to **char**, **varchar**, or **text** on the server. * **char**, **varchar**, or **text** server data sent to a Unicode SQL_C_WCHAR variable on the client. * ANSI SQL_C_CHAR client data sent to Unicode **nchar**, **nvarchar**, or **ntext** on the server. * Unicode **nchar**, **nvarchar**, or **ntext** server data sent to an ANSI SQL_C_CHAR variable on the client. When "no", character translation is not performed. The SQL Server Native Client ODBC driver does not translate client ANSI character SQL_C_CHAR data sent to **char**, **varchar**, or **text** variables, parameters, or columns on the server. No translation is performed on **char**, **varchar**, or **text** data sent from the server to SQL_C_CHAR variables on the client. If the client and SQL Server are using different ACPs, extended characters may be misinterpreted.
Database	Name of the default SQL Server database for the connection. If **Database** is not specified, the default database defined for the login is used. The default database from the ODBC data source overrides the default database defined for the login. The database must be an existing database unless **AttachDBFileName** is also specified. If **AttachDBFileName** is also specified, the primary file it points to is attached and given the database name specified by **Database**.
Driver	Name of the driver as returned by SQLDrivers. The keyword value for the SQL Server Native Client ODBC driver is "{SQL Server Native Client 11.0}". The **Server** keyword is required if **Driver** is specified and **DriverCompletion** is set to SQL_DRIVER_NOPROMPT. For more information about driver names, see Using the SQL Server Native Client Header and Library Files.

Keyword	Description
DSN	Name of an existing ODBC user or system data source. This keyword overrides any values that might be specified in the **Server**, **Network**, and **Address** keywords.
Encrypt	Specifies whether data should be encrypted before sending it over the network. Possible values are "yes"/"mandatory"(18.0+), "no"/"optional"(18.0+), and "strict" (18.0+). The default value is "yes" in version 18.0+ and "no" in previous versions.
Fallback	This keyword is deprecated, and its setting is ignored by the SQL Server Native Client ODBC driver.
Failover_Partner	Name of the failover partner server to be used if a connection cannot be made to the primary server.
FailoverPartnerSPN	The SPN for the failover partner. The default value is an empty string. An empty string causes SQL Server Native Client to use the default, driver-generated SPN.
FileDSN	Name of an existing ODBC file data source.
Language	SQL Server language name (optional). SQL Server can store messages for multiple languages in **sysmessages**. If connecting to a SQL Server with multiple languages, **Language** specifies which set of messages are used for the connection.
MARS_Connection	Enables or disables multiple active result sets (MARS) on the connection. Recognized values are "yes" and "no". The default is "no".
MultiSubnetFailover	Always specify **multiSubnetFailover=Yes** when connecting to the availability group listener of a SQL Server availability group or a SQL Server Failover Cluster Instance. **multiSubnetFailover=Yes** configures SQL Server Native Client to provide faster detection of and connection to the (currently) active server. Possible values are **Yes** and **No**. The default is **No**. For example: `MultiSubnetFailover=Yes` For more information about SQL Server Native Client's support for Always On availability groups, see SQL Server Native Client Support for High Availability, Disaster Recovery.
Net	Synonym for "Network".
Network	Valid values are **dbnmpntw** (named pipes) and **dbmssocn** (TCP/IP). It is an error to specify both a value for the **Network** keyword and a protocol prefix on the **Server** keyword.
PWD	The password for the SQL Server login account specified in the UID parameter. **PWD** need not be specified if the login has a NULL password or when using Windows Authentication (`Trusted_Connection = yes`).

Keyword	Description
QueryLog_On	When "yes", logging long-running query data is enabled on the connection. When "no", long-running query data is not logged.
QueryLogFile	Full path and file name of a file to use to log data on long-running queries.
QueryLogTime	Digit character string specifying the threshold (in milliseconds) for logging long-running queries. Any query that does not get a response in the time specified is written to the long-running query log file.
QuotedId	When "yes", QUOTED_IDENTIFIER is set ON for the connection, SQL Server uses the ISO rules regarding the use of quotation marks in SQL statements. When no, QUOTED_IDENTIFIER is set OFF for the connection. SQL Server then follows the legacy Transact-SQL rules regarding the use of quotation marks in SQL statements. For more information, see Effects of ISO Options.
Regional	When "yes", the SQL Server Native Client ODBC driver uses client settings when converting currency, date, and time data to character data. The conversion is one way only; the driver does not recognize non-ODBC standard formats for date strings or currency values within; for example, a parameter used in an INSERT or UPDATE statement. When "no", the driver uses ODBC standard strings to represent currency, date, and time data that is converted to character data.
SaveFile	Name of an ODBC data source file into which the attributes of the current connection are saved if the connection is successful.
Server	The name of a SQL Server instance. The value must be either the name of a server on the network, an IP address, or the name of a SQL Server Configuration Manager alias. The **Address** keyword overrides the **Server** keyword. You can connect to the default instance on the local server by specifying one of the following: **Server=;** **Server=.;** **Server=(local);** **Server=(local);** **Server=(localhost);** **Server=(localdb)** *instancename* **;** For more information about LocalDB support, see SQL Server Native Client Support for LocalDB.

Keyword	Description
	To specify a named instance of SQL Server, append *****InstanceName*. When no server is specified, a connection is made to the default instance on the local computer. If you specify an IP address, make sure that the TCP/IP or named pipes protocols are enabled in SQL Server Configuration Manager. The complete syntax for the **Server** keyword is as follows: **Server**=[*protocol***:**]*Server*[**,***port*] *protocol* can be **tcp** (TCP/IP), **lpc** (shared memory), or **np** (named pipes). The following is an example of specifying a named pipe: `np:\\.\pipe\MSSQL$MYINST01\sql\query` This line specifies named pipe protocol, a named pipe on the local machine (`\\.\pipe`), the name of the SQL Server instance (`MSSQL$MYINST01`), and the default name of the named pipe (`sql/query`). If neither a *protocol* nor the **Network** keyword is specified, SQL Server Native Client will use the protocol order specified in SQL Server Configuration Manager. *port* is the port to connect to, on the specified server. By default, SQL Server uses port 1433. Spaces are ignored at the beginning of the value passed to **Server** in ODBC connection strings when using SQL Server Native Client.
ServerSPN	The SPN for the server. The default value is an empty string. An empty string causes SQL Server Native Client to use the default, driver-generated SPN.
StatsLog_On	When "yes", enables the capture of SQL Server Native Client ODBC driver performance data. When "no", SQL Server Native Client ODBC driver performance data is not available on the connection.
StatsLogFile	Full path and file name of a file used to record SQL Server Native Client ODBC driver performance statistics.
Trusted_Connection	When "yes", instructs the SQL Server Native Client ODBC driver to use Windows Authentication Mode for login validation. Otherwise instructs the SQL Server Native Client ODBC driver to use a SQL Server username and password for login validation, and the UID and PWD keywords must be specified.

Keyword	Description
TrustServerCertificate	When used with **Encrypt**, enables encryption using a self-signed server certificate.
UID	A valid SQL Server login account. UID need not be specified when using Windows Authentication.
UseProcForPrepare	This keyword is deprecated, and its setting is ignored by the SQL Server Native Client ODBC Driver.
WSID	The workstation ID. Typically, this is the network name of the computer on which the application resides (optional). If specified, this value is stored in the **master.dbo.sysprocesses** column **hostname** and is returned by sp_who and the HOST_NAME function.

> **Note**
>
> Regional conversion settings apply to currency, numeric, date, and time data types. The conversion setting is only applicable to output conversion and is only visible when currency, numeric, date, or time values are converted to character strings.

The SQL Server Native Client ODBC driver uses the locale registry settings for the current user. The driver does not honor the current thread's locale if the application sets it after connection by, for example, calling **SetThreadLocale**.

Altering the regional behavior of a data source can cause application failure. An application that parses date strings, and expects date strings to appear as defined by ODBC, could be adversely affected by altering this value.

OLE DB Provider Connection String Keywords

There are two ways OLE DB applications can initialize data source objects:

- **IDBInitialize::Initialize**
- **IDataInitialize::GetDataSource**

In the first case, a provider string can be used to initialize connection properties by setting the property DBPROP_INIT_PROVIDERSTRING in the DBPROPSET_DBINIT property set. In the second case, an initialization string can be passed to **IDataInitialize::GetDataSource** method to initialize connection properties. Both methods initialize the same OLE DB connection properties, but different sets of keywords are used. The set of keywords used by

IDataInitialize::GetDataSource is at minimum the description of properties within the initialization property group.

Any provider string setting that has a corresponding OLE DB property set to some default value or explicitly set to a value, the OLE DB property value will override the setting in the provider string.

Boolean properties set in provider strings via DBPROP_INIT_PROVIDERSTRING values are set using the values "yes" and "no". Boolean properties set in initialization strings using **IDataInitialize::GetDataSource** are set using the values "true" and "false".

Applications using **IDataInitialize::GetDataSource** can also use the keywords used by **IDBInitialize::Initialize** but only for properties that do not have a default value. If an application uses both the **IDataInitialize::GetDataSource** keyword and the **IDBInitialize::Initialize** keyword in the initialization string, the **IDataInitialize::GetDataSource** keyword setting is used. It is strongly recommended that applications do not use **IDBInitialize::Initialize** keywords in **IDataInitialize:GetDataSource** connection strings, as this behavior may not be maintained in future releases.

> ⓘ **Note**
>
> A connection string passed through **IDataInitialize::GetDataSource** is converted into properties and applied via **IDBProperties::SetProperties**. If component services found the property description in **IDBProperties::GetPropertyInfo**, this property will be applied as a stand-alone property. Otherwise, it will be applied through DBPROP_PROVIDERSTRING property. For example, if you specify connection string **Data Source=server1;Server=server2**, **Data Source** will be set as a property, but **Server** will go into a provider string.

If you specify multiple instances of the same provider-specific property, the first value of the first property will be used.

Connection strings used by OLE DB applications using DBPROP_INIT_PROVIDERSTRING with **IDBInitialize::Initialize** have the following syntax:

```
connection-string ::= empty-string[;] | attribute[;] | attribute; connection-string
```

```
empty-string ::=
```

```
attribute ::= attribute-keyword=[{]attribute-value[}]
```

```
attribute-value ::= character-string
```

```
attribute-keyword ::= identifier
```

Attribute values can optionally be enclosed in braces, and it is good practice to do so. This avoids problems when attribute values contain non-alphanumeric characters. The first closing brace in the value is assumed to terminate the value, so values cannot contain closing brace characters.

A space character after the = sign of a connection string keyword will be interpreted as a literal, even if the value is enclosed in quotation marks.

The following table describes the keywords that may be used with DBPROP_INIT_PROVIDERSTRING.

Expand table

Keyword	Initialization property	Description
Addr	SSPROP_INIT_NETWORKADDRESS	Synonym for "Address".
Address	SSPROP_INIT_NETWORKADDRESS	The network address of an instance of SQL Server in the organization. For more information about valid address syntax, see the description of the **Address** ODBC keyword, later in this topic.
APP	SSPROP_INIT_APPNAME	The string identifying the application.
ApplicationIntent	SSPROP_INIT_APPLICATIONINTENT	Declares the application workload type when connecting to a server. Possible values are **ReadOnly** and **ReadWrite**. The default is **ReadWrite**. For more information about SQL Server Native Client's support for Always On availability groups, see SQL Server Native Client Support for High Availability, Disaster Recovery.
AttachDBFileName	SSPROP_INIT_FILENAME	The name of the primary file (include the full path name) of an attachable database. To

Keyword	Initialization property	Description
		use **AttachDBFileName**, you must also specify the database name with the provider string Database keyword. If the database was previously attached, SQL Server does not reattach it (it uses the attached database as the default for the connection).
Auto Translate	SSPROP_INIT_AUTOTRANSLATE	Synonym for "AutoTranslate".
AutoTranslate	SSPROP_INIT_AUTOTRANSLATE	Configures OEM/ANSI character translation. Recognized values are "yes" and "no".
Database	DBPROP_INIT_CATALOG	The database name.
DataTypeCompatibility	SSPROP_INIT_DATATYPECOMPATIBILITY	Specifies the mode of data type handling to use. Recognized values are "0" for provider data types and "80" for SQL Server 2000 data types.
Encrypt	SSPROP_INIT_ENCRYPT	Specifies whether data should be encrypted before sending it over the network. Possible values are "yes" and "no". The default value is "no".
FailoverPartner	SSPROP_INIT_FAILOVERPARTNER	The name of the failover server used for database mirroring.
FailoverPartnerSPN	SSPROP_INIT_FAILOVERPARTNERSPN	The SPN for the failover partner. The default value is an empty string. An empty string causes SQL Server Native Client to use the default, provider-generated SPN.
Language	SSPROP_INIT_CURRENTLANGUAGE	The SQL Server language.
MarsConn	SSPROP_INIT_MARSCONNECTION	Enables or disables multiple active result sets (MARS) on

Keyword	Initialization property	Description
		the connection if the server is SQL Server 2005 (9.x) or later. Possible values are "yes" and "no". The default value is "no".
Net	SSPROP_INIT_NETWORKLIBRARY	Synonym for "Network".
Network	SSPROP_INIT_NETWORKLIBRARY	The network library used to establish a connection to an instance of SQL Server in the organization.
Network Library	SSPROP_INIT_NETWORKLIBRARY	Synonym for "Network".
PacketSize	SSPROP_INIT_PACKETSIZE	Network packet size. The default is 4096.
PersistSensitive	DBPROP_AUTH_PERSIST_SENSITIVE_AUTHINFO	Accepts the strings "yes" and "no" as values. When "no", the data source object is not allowed to persist sensitive authentication information
PWD	DBPROP_AUTH_PASSWORD	The SQL Server login password.
Server	DBPROP_INIT_DATASOURCE	The name of an instance of SQL Server in the organization. When not specified, a connection is made to the default instance on the local computer. For more information about valid address syntax, see the description of the **Server** ODBC keyword, in this topic.
ServerSPN	SSPROP_INIT_SERVERSPN	The SPN for the server. The default value is an empty string. An empty string causes SQL Server Native Client to use the default, provider-generated SPN.

Keyword	Initialization property	Description
Timeout	DBPROP_INIT_TIMEOUT	The amount of time (in seconds) to wait for data source initialization to complete.
Trusted_Connection	DBPROP_AUTH_INTEGRATED	When "yes", instructs the SQL Server Native Client OLE DB provider to use Windows Authentication Mode for login validation. Otherwise instructs the SQL Server Native Client OLE DB provider to use a SQL Server username and password for login validation, and the UID and PWD keywords must be specified.
TrustServerCertificate	SSPROP_INIT_TRUST_SERVER_CERTIFICATE	Accepts the strings "yes" and "no" as values. The default value is "no", which means that the server certificate will be validated.
UID	DBPROP_AUTH_USERID	The SQL Server login name.
UseProcForPrepare	SSPROP_INIT_USEPROCFORPREP	This keyword is deprecated, and its setting is ignored by the SQL Server Native Client OLE DB Provider.
WSID	SSPROP_INIT_WSID	The workstation identifier.

Connection strings used by OLE DB applications using **IDataInitialize::GetDataSource** have the following syntax:

```
connection-string ::= empty-string[;] | attribute[;] | attribute; connection-string
```

```
empty-string ::=
```

```
attribute ::= attribute-keyword=[quote]attribute-value[quote]
```

```
attribute-value ::= character-string
```

```
attribute-keyword ::= identifier
```

```
quote ::= " | '
```

Property use must conform to the syntax permitted in its scope. For example, **WSID** uses curly braces ({}) quotation characters and **Application Name** uses single (') or double (") quotation characters. Only string properties can be quoted. Attempting to quote an integer or enumerated property will result in a "Connection String does not conform to OLE DB specification" error.

Attribute values can optionally be enclosed in single or double quotes, and it is good practice to do so. This avoids problems when values contain non-alphanumeric characters. The quote character used can also appear in values, provided that it is doubled.

A space character after the = sign of a connection string keyword will be interpreted as a literal, even if the value is enclosed in quotation marks.

If a connection string has more than one of the properties listed in the following table, the value of the last property will be used.

The following table describes the keywords that may be used with **IDataInitialize::GetDataSource**:

Expand table

Keyword	Initialization property	Description
Application Name	SSPROP_INIT_APPNAME	The string identifying the application.
Application Intent	SSPROP_INIT_APPLICATIONINTENT	Synonym for "ApplicationIntent".
ApplicationIntent	SSPROP_INIT_APPLICATIONINTENT	Declares the application workload type when connecting to a server. Possible values are **ReadOnly** and **ReadWrite**. The default is **ReadWrite**. For more information about SQL Server Native Client's support for Always On availability groups, see SQL Server Native Client Support for High Availability, Disaster Recovery.
Auto Translate	SSPROP_INIT_AUTOTRANSLATE	Synonym for "AutoTranslate".
AutoTranslate	SSPROP_INIT_AUTOTRANSLATE	Configures OEM/ANSI character translation.

Keyword	Initialization property	Description
		Recognized values are "true" and "false".
Connect Timeout	DBPROP_INIT_TIMEOUT	The amount of time (in seconds) to wait for data source initialization to complete.
Current Language	SSPROP_INIT_CURRENTLANGUAGE	The SQL Server language name.
Data Source	DBPROP_INIT_DATASOURCE	The name of an instance of SQL Server in the organization. When not specified, a connection is made to the default instance on the local computer. For more information about valid address syntax, see the description of the **Server** ODBC keyword, later in this topic.
DataTypeCompatibility	SSPROP_INIT_DATATYPECOMPATIBILITY	Specifies the mode of data type handling to use. Recognized values are "0" for provider data types and "80" for SQL Server 2000 (8.x) data types.
Failover Partner	SSPROP_INIT_FAILOVERPARTNER	The name of the failover server used for database mirroring.
Failover Partner SPN	SSPROP_INIT_FAILOVERPARTNERSPN	The SPN for the failover partner. The default value is an empty string. An empty string causes SQL Server Native Client to use the default, provider-generated SPN.
Initial Catalog	DBPROP_INIT_CATALOG	The database name.
Initial File Name	SSPROP_INIT_FILENAME	The name of the primary file (include the full path name)

Keyword	Initialization property	Description
		of an attachable database. To use **AttachDBFileName**, you must also specify the database name with the provider string DATABASE keyword. If the database was previously attached, SQL Server does not reattach it (it uses the attached database as the default for the connection).
Integrated Security	DBPROP_AUTH_INTEGRATED	Accepts the value "SSPI" for Windows Authentication.
MARS Connection	SSPROP_INIT_MARSCONNECTION	Enables or disables multiple active result sets (MARS) on the connection. Recognized values are "true" and "false". The default is "false".
Network Address	SSPROP_INIT_NETWORKADDRESS	The network address of an instance of SQL Server in the organization. For more information about valid address syntax, see the description of the **Address** ODBC keyword, later in this topic.
Network Library	SSPROP_INIT_NETWORKLIBRARY	The network library used to establish a connection to an instance of SQL Server in the organization.
Packet Size	SSPROP_INIT_PACKETSIZE	Network packet size. The default is 4096.
Password	DBPROP_AUTH_PASSWORD	The SQL Server login password.
Persist Security Info	DBPROP_AUTH_PERSIST_SENSITIVE_AUTHINFO	Accepts the strings "true" and "false" as values. When "false", the data source object is not allowed to persist sensitive authentication information

Keyword	Initialization property	Description
Provider		For SQL Server Native Client, this should be "SQLNCLI11".
Server SPN	SSPROP_INIT_SERVERSPN	The SPN for the server. The default value is an empty string. An empty string causes SQL Server Native Client to use the default, provider-generated SPN.
Trust Server Certificate	SSPROP_INIT_TRUST_SERVER_CERTIFICATE	Accepts the strings "true" and "false" as values. The default value is "false", which means that the server certificate will be validated.
Use Encryption for Data	SSPROP_INIT_ENCRYPT	Specifies whether data should be encrypted before sending it over the network. Possible values are "true" and "false". The default value is "false".
User ID	DBPROP_AUTH_USERID	The SQL Server login name.
Workstation ID	SSPROP_INIT_WSID	The workstation identifier.

Note In the connection string, the "Old Password" property sets SSPROP_AUTH_OLD_PASSWORD, which is the current (possibly expired) password that is not available via a provider string property.

ActiveX Data Objects (ADO) Connection String Keywords

ADO applications set the **ConnectionString** property of **ADODBConnection** objects or supply a connection string as a parameter to the **Open** method of **ADODBConnection** objects.

ADO applications can also use the keywords used by the OLE DB **IDBInitialize::Initialize** method, but only for properties that do not have a default value. If an application uses both the ADO keywords and the **IDBInitialize::Initialize** keywords in the initialization string, the ADO keyword setting will be used. It is strongly recommended that applications only use ADO connection string keywords.

Connection strings used by ADO have the following syntax:

```
connection-string ::= empty-string[;] | attribute[;] | attribute; connection-string

empty-string ::=

attribute ::= attribute-keyword=["]attribute-value["]

attribute-value ::= character-string

attribute-keyword ::= identifier
```

Attribute values can optionally be enclosed in double quotes, and it is good practice to do so. This avoids problems when values contain non-alphanumeric characters. Attribute values cannot contain double quotes.

The following table describes the keywords that may be used with an ADO connection string:

Expand table

Keyword	Initialization property	Description
Application Intent	SSPROP_INIT_APPLICATIONINTENT	Synonym for "ApplicationIntent".
ApplicationIntent	SSPROP_INIT_APPLICATIONINTENT	Declares the application workload type when connecting to a server. Possible values are **ReadOnly** and **ReadWrite**. The default is **ReadWrite**. For more information about SQL Server Native Client's support for Always On availability groups, see SQL Server Native Client Support for High Availability, Disaster Recovery.
Application Name	SSPROP_INIT_APPNAME	The string identifying the application.
Auto Translate	SSPROP_INIT_AUTOTRANSLATE	Synonym for "AutoTranslate".
AutoTranslate	SSPROP_INIT_AUTOTRANSLATE	Configures OEM/ANSI character translation. Recognized values are "true" and "false".
Connect Timeout	DBPROP_INIT_TIMEOUT	The amount of time (in seconds) to wait for data

Keyword	Initialization property	Description
		source initialization to complete.
Current Language	SSPROP_INIT_CURRENTLANGUAGE	The SQL Server language name.
Data Source	DBPROP_INIT_DATASOURCE	The name of an instance of SQL Server in the organization. When not specified, a connection is made to the default instance on the local computer. For more information about valid address syntax, see the description of the **Server** ODBC keyword, in this topic.
DataTypeCompatibility	SSPROP_INIT_DATATYPECOMPATIBILITY	Specifies the mode of data type handling that will be used. Recognized values are "0" for provider data types and "80" for SQL Server 2000 data types.
Failover Partner	SSPROP_INIT_FAILOVERPARTNER	The name of the failover server used for database mirroring.
Failover Partner SPN	SSPROP_INIT_FAILOVERPARTNERSPN	The SPN for the failover partner. The default value is an empty string. An empty string causes SQL Server Native Client to use the default, provider-generated SPN.
Initial Catalog	DBPROP_INIT_CATALOG	The database name.
Initial File Name	SSPROP_INIT_FILENAME	The name of the primary file (include the full path name) of an attachable database. To use **AttachDBFileName**, you must also specify the database name with the provider string DATABASE keyword. If the database was

Keyword	Initialization property	Description
		previously attached, SQL Server does not reattach it (it uses the attached database as the default for the connection).
Integrated Security	DBPROP_AUTH_INTEGRATED	Accepts the value "SSPI" for Windows Authentication.
MARS Connection	SSPROP_INIT_MARSCONNECTION	Enables or disables multiple active result sets (MARS) on the connection if the server is SQL Server 2005 (9.x) or later. Recognized values are "true" and "false".The default is "false".
Network Address	SSPROP_INIT_NETWORKADDRESS	The network address of an instance of SQL Server in the organization. For more information about valid address syntax, see the description of the **Address** ODBC keyword, in this topic.
Network Library	SSPROP_INIT_NETWORKLIBRARY	The network library used to establish a connection to an instance of SQL Server in the organization.
Packet Size	SSPROP_INIT_PACKETSIZE	Network packet size. The default is 4096.
Password	DBPROP_AUTH_PASSWORD	The SQL Server login password.
Persist Security Info	DBPROP_AUTH_PERSIST_SENSITIVE_AUTHINFO	Accepts the strings "true" and "false" as values. When "false" the data source object is not allowed to persist sensitive authentication information.
Provider		For SQL Server Native Client, this should be "SQLNCLI11".
Server SPN	SSPROP_INIT_SERVERSPN	The SPN for the server. The default value is an empty string. An empty string causes

Keyword	Initialization property	Description
		SQL Server Native Client to use the default, provider-generated SPN.
Trust Server Certificate	SSPROP_INIT_TRUST_SERVER_CERTIFICATE	Accepts the strings "true" and "false" as values. The default value is "false", which means that the server certificate will be validated.
Use Encryption for Data	SSPROP_INIT_ENCRYPT	Specifies whether data should be encrypted before sending it over the network. Possible values are "true" and "false". The default value is "false".
User ID	DBPROP_AUTH_USERID	The SQL Server login name.
Workstation ID	SSPROP_INIT_WSID	The workstation identifier.

> **Note**
>
> In the connection string, the "Old Password" property sets SSPROP_AUTH_OLD_PASSWORD, which is the current (possibly expired) password that is not available via a provider string property.

See also

Building Applications with SQL Server Native Client

SQL Server Browser service

Article • 02/13/2025

Applies to: SQL Server - Windows only

The SQL Server Browser program runs as a Windows service. SQL Server Browser listens for incoming requests for SQL Server resources and provides information about SQL Server instances installed on the computer. SQL Server Browser contributes to the following actions:

- Browsing a list of available servers.
- Connecting to the correct server instance.
- Connecting to dedicated administrator connection (DAC) endpoints.

For each instance of the Database Engine and SQL Server Analysis Services (SSAS), the SQL Server Browser service (sqlbrowser) provides the instance name and the version number. SQL Server Browser is installed with SQL Server.

SQL Server Browser can be configured during setup or using SQL Server Configuration Manager. By default, the SQL Server Browser service starts automatically:

- When upgrading an installation.
- When installing on a cluster.
- When installing a named instance of the Database Engine, including all instances of SQL Server Express.
- When installing a named instance of SSAS.

Background

Before SQL Server 2000 (8.x), only one instance of SQL Server could be installed on a computer. SQL Server listened for incoming requests on port 1433, assigned to SQL Server by the official Internet Assigned Numbers Authority (IANA). Only one instance of SQL Server can use a port, so when SQL Server 2000 (8.x) introduced support for multiple instances of SQL Server, SQL Server Resolution Protocol (SSRP) was developed to listen on User Datagram Protocol (UDP) port 1434. This listener service responds to client requests with the names of the installed instances and the ports or named pipes used by the instance.

To resolve limitations of the SSRP system, SQL Server 2005 (9.x) introduced the SQL Server Browser service as a replacement for SSRP.

How SQL Server Browser works

When an instance of SQL Server starts and the TCP/IP protocol is enabled for SQL Server, the server is assigned a TCP/IP port. SQL Server listens on a specific named pipe if the named pipes protocol is enabled. This port or named pipe is used by that specific instance to exchange data with client applications. TCP/IP port 1433 and pipe `\sql\query` are assigned to the default instance during installation. The server administrator can change the port or named pipe using SQL Server Configuration Manager.

Because only one instance of SQL Server can use a port or pipe, different port numbers and pipe names are assigned for named instances, including SQL Server Express. When enabled, named instances and SQL Server Express are configured to use dynamic ports by default. That is, an available port is assigned when SQL Server starts.

If you want, a specific port can be assigned to an instance of SQL Server. Clients can specify a specific port when connecting to SQL Server. However, if the port is dynamically assigned, the port number can change anytime the SQL Server is restarted, so the correct port number is unknown to the client.

Upon startup, the SQL Server Browser starts and claims UDP port 1434. SQL Server Browser reads the registry, identifies all instances of SQL Server on the computer, and notes the ports and named pipes they use. When a server has two or more network cards, SQL Server Browser returns the first enabled port it encounters for SQL Server. SQL Server Browser supports ipv6 and ipv4.

When SQL Server clients request SQL Server resources, the client network library sends a UDP message to the server using port 1434. SQL Server Browser responds with the TCP/IP port or named pipe of the requested instance. The network library on the client application then completes the connection by sending a request to the server using the port or named pipe of the desired instance.

For more information on starting and stopping the SQL Server Browser service, see Start, stop, pause, resume, and restart SQL Server services.

Use SQL Server Browser

If the SQL Server Browser service isn't running, you can still connect to the SQL Server if you provide the correct port number or named pipe. For instance, you can connect to the default instance of SQL Server with TCP/IP if it's running on port 1433.

However, if the SQL Server Browser service isn't running, the following connections don't work:

- Any component that tries to connect to a named instance without fully specifying all the parameters (such as the TCP/IP port or named pipe).

- Any component that generates or passes `<server>\<instance>` information that other components could later use to reconnect.
- Connecting to a named instance without providing the port number or pipe.
- Diagnostic connection for database administrators to a named instance or the default instance if not using TCP/IP port 1433.
- The Online Analytical Processing (OLAP) redirector service.
- Enumerating servers in SQL Server Management Studio.

Suppose you're using SQL Server in a client-server scenario (for example, when your application is accessing SQL Server across a network). If you stop or disable the SQL Server Browser service, you must assign a specific port number to each instance and configure your client application code to use that port number. This approach has the following problems:

- You must update and maintain client application code to ensure it's connecting to the proper port.
- The port you choose for each instance could be used by another service or application on the server, causing the instance of SQL Server to be unavailable.

Clusters and SQL Server Browser

SQL Server Browser isn't a clustered resource and doesn't support failover from one cluster node to the other. Therefore, if there's a cluster, SQL Server Browser should be installed and turned on for each cluster node. On clusters, the SQL Server Browser listens on `IP_ANY`.

When the SQL Server Browser listens on `IP_ANY` and you enable listening on specific IP addresses, you must configure the same TCP/IP port on each IP address, because SQL Server Browser returns the first IP address and port pair it encounters.

Install, uninstall, and run from the command line

By default, the SQL Server Browser program is installed at `<drive>:\Program Files (x86)\Microsoft SQL Server\<nn>\Shared\sqlbrowser.exe`.

The SQL Server Browser service is uninstalled when the last instance of SQL Server is removed.

SQL Server Browser can be started from the command prompt for troubleshooting by using the `-c` switch:

```
```

Windows Command Prompt

```
<drive>\<path>\sqlbrowser.exe -c
```

Security

The SQL Server Browser Service is crucial in facilitating network communication with SQL Server instances.

Security measures for SQL Server Browser Service include:

- Configuring firewalls to allow its traffic.
- Restricting access to trusted IP addresses.
- Regularly applying updates to patch vulnerabilities.
- Additionally, you must implement strong authentication and authorization policies to prevent unauthorized access and maintain the integrity of your SQL Server environment.

Account privileges

SQL Server Browser listens on a UDP port and accepts unauthenticated requests using SQL Server Resolution Protocol (SSRP). SQL Server Browser should be run in the security context of a low-privileged user to minimize exposure to a malicious attack. The sign in account can be changed by using the SQL Server Configuration Manager.

The minimum user rights for SQL Server Browser are:

- Deny access to this computer from the network.
- Deny sign in locally.
- Deny sign in as a batch job.
- Deny Log On Through Terminal Services.
- Sign in as a service.
- Read and write the SQL Server registry keys related to network communication (ports and pipes).

Default account

Setup configures SQL Server Browser to use the account selected for services during setup. Other possible accounts include:

- Any **domain\local** account.

- The **Local Service** account.
- The **Local System** account (not recommended as it has unnecessary privileges).

Hide SQL Server

Hidden instances are instances of SQL Server that support only shared memory connections. For SQL Server, set the `HideInstance` flag to indicate that SQL Server Browser shouldn't respond with information about this server instance.

Use a firewall

To communicate with the SQL Server Browser service on a server behind a firewall, open UDP port 1434 and the TCP/IP port used by SQL Server (for example, 1433). For information about working with a firewall, see Configure the Windows Firewall to Allow SQL Server Access.

Related content

- Network Protocols and Network Libraries
- Connecting Using IPv6
- Enable or disable a server network protocol

Choose an authentication mode

Article • 09/16/2024

Applies to: SQL Server

During setup, you must select an authentication mode for the Database Engine. There are two possible modes: Windows Authentication mode and mixed mode. Windows Authentication mode enables Windows Authentication and disables SQL Server Authentication. Mixed mode enables both Windows Authentication and SQL Server Authentication. Windows Authentication is always available and can't be disabled.

Configuring the authentication mode

If you select Mixed Mode Authentication (SQL Server and Windows Authentication mode) during setup, you must provide and then confirm a strong password for the built-in SQL Server system administrator account named `sa`. The `sa` account connects by using SQL Server Authentication.

If you select Windows Authentication during setup, the setup creates the `sa` account for SQL Server authentication but it's disabled. If you later change to Mixed Mode Authentication and you want to use the `sa` account, you must enable the account. Any Windows or SQL Server account can be configured as a system administrator. Because the `sa` account is well known and often targeted by malicious users, don't enable the `sa` account unless your application requires it. Never set a blank or weak password for the `sa` account. To change from Windows Authentication mode to Mixed Mode Authentication and use SQL Server Authentication, see Change Server Authentication Mode.

Connecting through Windows Authentication

When a user connects through a Windows user account, SQL Server validates the account name and password using the Windows principal token in the operating system. This means that the user identity is confirmed by Windows. SQL Server doesn't ask for the password, and doesn't perform the identity validation. Windows Authentication is the default authentication mode, and is much more secure than SQL Server Authentication. Windows Authentication uses New Technology LAN Manager (NTLM) or Kerberos security protocol, provides password policy enforcement with regard to complexity validation for strong passwords, provides support for account lockout, and supports password expiration. A connection made using Windows Authentication is sometimes called a trusted connection, because SQL Server trusts the credentials provided by Windows.

For information on configuring Kerberos, see Register a Service Principal Name for Kerberos connections.

By using Windows Authentication, Windows groups can be created at the domain level, and a login can be created on SQL Server for the entire group. Managing access at the domain level can simplify account administration.

> **Important**
>
> When possible, use Windows authentication.

Connecting through SQL Server Authentication

When using SQL Server Authentication, logins are created in SQL Server that aren't based on Windows user accounts. Both the user name and the password are created by using SQL Server and stored in SQL Server. Users connecting using SQL Server Authentication must provide their credentials (login and password) every time that they connect. When using SQL Server Authentication, you must set strong passwords for all SQL Server accounts. For strong password guidelines, see Strong Passwords.

Three optional password policies are available for SQL Server logins.

- User must change password at next login

  Requires the user to change the password the next time that the user connects. The ability to change the password is provided by SQL Server Management Studio. Other company software developers should provide this feature if this option is used.

- Enforce password expiration

  The maximum password age policy of the computer is enforced for SQL Server logins.

- Enforce password policy

  The Windows password policies of the computer are enforced for SQL Server logins. This includes password length and complexity. This functionality depends on the `NetValidatePasswordPolicy` API, which is only available in Windows Server 2003 and later versions.

To determine the password policies of the local computer

1. On the **Start** menu, select **Run**.

2. In the **Run** dialog box, type **secpol.msc**, and then select **OK**.
3. In the **Local Security Settings** application, expand **Security Settings**, expand **Account Policies**, and then select **Password Policy**.

   The password policies are described in the results pane.

Disadvantages of SQL Server Authentication

- If a user is a Windows domain user who has a login and password for Windows, they must still provide another (SQL Server) login and password to connect. Keeping track of multiple names and passwords is difficult for many users. Having to provide SQL Server credentials every time the user connects to the database can be annoying.
- SQL Server Authentication can't use Kerberos security protocol.
- Windows offers additional password policies that aren't available for SQL Server logins.
- The encrypted SQL Server Authentication login password must be passed over the network at the time of the connection. Some applications that connect automatically will store the password at the client. These are additional attack points.

Advantages of SQL Server Authentication

- Allows SQL Server to support older applications and applications provided by third parties that require SQL Server Authentication.
- Allows SQL Server to support environments with mixed operating systems, where all users aren't authenticated by a Windows domain.
- Allows users to connect from unknown or untrusted domains. For instance, an application where established customers connect with assigned SQL Server logins to receive the status of their orders.
- Allows SQL Server to support web-based applications where users create their own identities.
- Allows software developers to distribute their applications by using a complex permission hierarchy based on known, preset SQL Server logins.

> **Note**

Using SQL Server Authentication does not limit the permissions of local administrators on the computer where SQL Server is installed.

Related content

- Security Considerations for a SQL Server Installation

Integration Services Service (SSIS Service)

Article • 12/17/2024

Applies to: SQL Server SSIS Integration Runtime in Azure Data Factory

The topics in this section discuss the Integration Services service, a Windows service for managing Integration Services packages. This service is not required to create, save, and run Integration Services packages. SQL Server 2012 (11.x) supports the Integration Services service for backward compatibility with earlier releases of Integration Services.

Starting in SQL Server 2012 (11.x), Integration Services stores objects, settings, and operational data in the **SSISDB** database for projects that you've deployed to the Integration Services server using the project deployment model. The Integration Services server, which is an instance of the SQL Server Database Engine, hosts the database. For more information about the database, see SSIS Catalog. For more information about deploying projects to the Integration Services server, see Deploy Integration Services (SSIS) Projects and Packages.

Management capabilities

The Integration Services service is a Windows service for managing Integration Services packages. The Integration Services service is available only in SQL Server Management Studio.

Running the Integration Services service provides the following management capabilities:

- Starting remote and locally stored packages
- Stopping remote and locally running packages
- Monitoring remote and locally running packages
- Importing and exporting packages
- Managing package storage
- Customizing storage folders
- Stopping running packages when the service is stopped
- Viewing the Windows Event log
- Connecting to multiple Integration Services servers

Startup type

The Integration Services service is installed when you install the Integration Services component of SQL Server. By default, the Integration Services service is started and the startup type of the service is set to automatic. The service must be running to monitor the packages that are stored in the SSIS Package Store. The SSIS Package Store can be either the msdb database in an instance of SQL Server or the designated folders in the file system.

The Integration Services service is not required if you only want to design and execute Integration Services packages. However, the service is required to list and monitor packages using SQL Server Management Studio.

Manage the service

When you install the Integration Services component of SQL Server, the Integration Services service is also installed. By default, the Integration Services service is started and the startup type of the service is set to automatic. However, you must also install SQL Server Management Studio to use the service to manage stored and running Integration Services packages.

> **Note**
>
> To connect directly to an instance of the legacy Integration Services Service, you have to use the version of SQL Server Management Studio (SSMS) aligned with the version of SQL Server on which the Integration Services Service is running. For example, to connect to the legacy Integration Services Service running on an instance of SQL Server 2016, you have to use the version of SSMS released for SQL Server 2016. **Download SQL Server Management Studio (SSMS)**.
>
> In the SSMS **Connect to Server** dialog box, you cannot enter the name of a server on which an earlier version of the Integration Services service is running. However, to manage packages that are stored on a remote server, you do not have to connect to the instance of the Integration Services service on that remote server. Instead, edit the configuration file for the Integration Services service so that SQL Server Management Studio displays the packages that are stored on the remote server.

You can only install a single instance of the Integration Services service on a computer. The service is not specific to a particular instance of the Database Engine. You connect to the service by using the name of the computer on which it is running.

You can manage the Integration Services service by using one of the following Microsoft Management Console (MMC) snap-ins: SQL Server Configuration Manager or Services. Before you can manage packages in SQL Server Management Studio, you must make sure that the service is started.

By default, the Integration Services service is configured to manage packages in the msdb database of the instance of the Database Engine that is installed at the same time as Integration Services. If an instance of the Database Engine is not installed at the same time, the Integration Services service is configured to manage packages in the msdb database of the local, default instance of the Database Engine. To manage packages that are stored in a named or remote instance of the Database Engine, or in multiple instances of the Database Engine, you have to modify the configuration file for the service.

By default, the Integration Services service is configured to stop running packages when the service is stopped. However, the Integration Services service does not wait for packages to stop and some packages may continue running after the Integration Services service is stopped.

If the Integration Services service is stopped, you can continue to run packages using the SQL Server Import and Export Wizard, the SSIS Designer, the Execute Package Utility, and the **dtexec** command prompt utility (dtexec.exe). However, you cannot monitor the running packages.

By default, the Integration Services service runs in the context of the NETWORK SERVICE account. It is recommended to run the SQL Server Integration Services service under an account that has limited permissions such as the NETWORK SERVICE account. Running the SQL Server Integration Services service under a highly-privileged account represents a potential security risk.

The Integration Services service writes to the Windows event log. You can view service events in SQL Server Management Studio. You can also view service events by using the Windows Event Viewer.

Set the properties of the service

The Integration Services service manages and monitors packages in SQL Server Management Studio. When you first install SQL Server Integration Services, the Integration Services service is started and the startup type of the service is set to automatic.

After the Integration Services service has been installed, you can set the properties of the service by using either SQL Server Configuration Manager or the Services MMC snap-in.

To configure other important features of the service, including the locations where it stores and manages packages, you must modify the configuration file of the service.

To set properties of the Integration Services service by using SQL Server Configuration Manager

1. On the **Start** menu, point to **All Programs**, point to **Microsoft SQL Server**, point to **Configuration Tools**, and then click **SQL Server Configuration Manager**.
2. In the **SQL Server Configuration Manager** snap-in, locate **SQL Server Integration Services** in the list of services, right-click **SQL Server Integration Services**, and then click **Properties**.
3. In the **SQL Server Integration Services Properties** dialog box you can do the following:
   - Click the **Log On** tab to view the logon information such as the account name.
   - Click the **Service** tab to view information about the service such as the name of the host computer and to specify the start mode of Integration Services service.

     > ⓘ **Note**
     >
     > The **Advanced** tab contains no information for Integration Services service.
4. Click **OK**.
5. On the **File** menu, click **Exit** to close the **SQL Server Configuration Manager** snap-in.

To set properties of the Integration Services service by using Services

1. In **Control Panel**, if you are using Classic View, click **Administrative Tools**, or, if you are using Category View, click **Performance and Maintenance** and then click **Administrative Tools**.
2. Click **Services**.
3. In the **Services** snap-in, locate **SQL Server Integration Services** in the list of services, right-click **SQL Server Integration Services**, and then click **Properties**.
4. In the **SQL Server Integration Services Properties** dialog box, you can do the following:
   - Click the **General** tab. To enable the service, select either the manual or automatic startup type. To disable the service, select Disable in the **Startup type** box. Selecting Disable does not stop the service if it is currently running.

     If the service is already enabled, you can click **Stop** to stop the service, or click **Start** to start the service.
   - Click the **Log On** tab to view or edit the logon information.

- Click the **Recovery** tab to view the default computer responses to service failure. You can modify these options to suit your environment.
- Click the **Dependencies** tab to view a list of dependent services. The Integration Services service has no dependencies.

5. Click **OK**.
6. Optionally, if the startup type is Manual or Automatic, you can right-click **SQL Server Integration Services** and click **Start, Stop, or Restart**.
7. On the **File** menu, click **Exit** to close the **Services** snap-in.

Grant permissions to the service

In previous versions of SQL Server, by default when you installed SQL Server all users in the Users group in Local Users and Groups had access to the Integration Services service. When you install SQL Server 2016 (13.x) and later versions, users do not have access to the Integration Services service. The service is secure by default. After SQL Server is installed, the administrator must grant access to the service.

To grant access to the Integration Services service

1. Run Dcomcnfg.exe. Dcomcnfg.exe provides a user interface for modifying certain settings in the registry.
2. In the **Component Services** dialog, expand the Component Services > Computers > My Computer > DCOM Config node.
3. Right-click **Microsoft SQL Server Integration Services 13.0**, and then click **Properties**.
4. On the **Security** tab, click **Edit** in the **Launch and Activation Permissions** area.
5. Add users and assign appropriate permissions, and then click Ok.
6. Repeat steps 4 - 5 for Access Permissions.
7. Restart SQL Server Management Studio.
8. Restart the Integration Services Service.

Event logged when permissions are missing

If the service account of the SQL Server Agent doesn't have the Integration Services DCOM **[Launch and Activation Permissions]**, the following event is added to the system event logs when the SQL Server Agent executes the SSIS package jobs:

```
Log Name: System
Source: **Microsoft-Windows-DistributedCOM**
Date: 1/9/2019 5:42:13 PM
Event ID: **10016**
Task Category: None
Level: Error
Keywords: Classic
User: NT SERVICE\SQLSERVERAGENT
Computer: testmachine
Description:
The application-specific permission settings do not grant Local Activation
permission for the COM Server application with CLSID
{xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx}
and APPID
{xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx}
to the user NT SERVICE\SQLSERVERAGENT SID (S-1-5-80-344959196-2060754871-
2302487193-2804545603-1466107430) from address LocalHost (Using LRPC) running in
the application container Unavailable SID (Unavailable). This security permission
can be modified using the Component Services administrative tool.
```

Configure the service

When you install Integration Services, the setup process creates and installs the configuration file for the Integration Services service. This configuration file contains the following settings:

- Packages are sent a stop command when the service stops.
- The root folders to display for Integration Services in Object Explorer of SQL Server Management Studio are the MSDB and File System folders.
- The packages in the file system that the Integration Services service manages are located in %ProgramFiles%\Microsoft SQL Server\130\DTS\Packages.

This configuration file also specifies which msdb database contains the packages that the Integration Services service will manage. By default, the Integration Services service is configured to manage packages in the msdb database of the instance of the Database Engine that is installed at the same time as Integration Services. If an instance of the Database Engine is not installed at the same time, the Integration Services service is configured to manage packages in the msdb database of the local, default instance of the Database Engine.

Default Configuration File Example

The following example shows a default configuration file that specifies the following settings:

- Packages stop running when the Integration Services service stops.
- The root folders for package storage in Integration Services are MSDB and File System.
- The service manages packages that are stored in the msdb database of the local, default instance of SQL Server.
- The service manages packages that are stored in the file system in the Packages folder.

Example of a Default Configuration File

XML

```
\<?xml version="1.0" encoding="utf-8"?>
\<DtsServiceConfiguration xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <StopExecutingPackagesOnShutdown>true</StopExecutingPackagesOnShutdown>
  <TopLevelFolders>
    \<Folder xsi:type="SqlServerFolder">
      <Name>MSDB</Name>
      <ServerName>.</ServerName>
    </Folder>
    \<Folder xsi:type="FileSystemFolder">
      <Name>File System</Name>
      <StorePath>..\Packages</StorePath>
    </Folder>
  </TopLevelFolders>
</DtsServiceConfiguration>
```

Modify the configuration file

You can modify the configuration file to allow packages to continue running if the service stops, to display additional root folders in Object Explorer, or to specify a different folder or additional folders in the file system to be managed by Integration Services service. For example, you can create additional root folders of type, **SqlServerFolder**, to manage packages in the msdb databases of additional instances of Database Engine.

> ⓘ **Note**
>
> Some characters are not valid in folder names. Valid characters for folder names are determined by the .NET Framework class **System.IO.Path** and the **GetInvalidFilenameChars** field. The **GetInvalidFilenameChars** field provides a platform-

specific array of characters that cannot be specified in path string arguments passed to members of the **Path** class. The set of invalid characters can vary by file system. Typically, invalid characters are the quotation mark ("), less than (<) character, and pipe (|) character.

However, you will have to modify the configuration file to manage packages that are stored in a named instance or a remote instance of Database Engine. If you do not update the configuration file, you cannot use **Object Explorer** in SQL Server Management Studio to view packages that are stored in the msdb database on the named instance or the remote instance. If you try to use **Object Explorer** to view these packages, you receive the following error message:

```
Failed to retrieve data for this request. (Microsoft.SqlServer.SmoEnum)
```

```
The SQL Server specified in Integration Services service configuration is not present or is not available. This might occur when there is no default instance of SQL Server on the computer. For more information, see the topic "Configuring the Integration Services Service" in SQL Server 2008 Books Online.
```

```
Login Timeout Expired
```

```
An error has occurred while establishing a connection to the server. When connecting to SQL Server 2008, this failure may be caused by the fact that under the default settings SQL Server does not allow remote connections.
```

```
Named Pipes Provider: Could not open a connection to SQL Server [2]. (MsDtsSvr).
```

To modify the configuration file for the Integration Services service, you use a text editor.

> **Important**
>
> After you modify the service configuration file, you must restart the service to use the updated service configuration.

Modified Configuration File Example

The following example shows a modified configuration file for Integration Services. This file is for a named instance of SQL Server called `InstanceName` on a server named `ServerName`.

Example of a Modified Configuration File for a Named Instance of SQL Server

XML

```
\<?xml version="1.0" encoding="utf-8"?>
\<DtsServiceConfiguration xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <StopExecutingPackagesOnShutdown>true</StopExecutingPackagesOnShutdown>
  <TopLevelFolders>
    \<Folder xsi:type="SqlServerFolder">
      <Name>MSDB</Name>
      <ServerName>ServerName\InstanceName</ServerName>
    </Folder>
    \<Folder xsi:type="FileSystemFolder">
      <Name>File System</Name>
      <StorePath>..\Packages</StorePath>
    </Folder>
  </TopLevelFolders>
</DtsServiceConfiguration>
```

Modify the Configuration File Location

The Registry key **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\130\SSIS\ServiceConfigFile** specifies the location and name for the configuration file that Integration Services service uses. The default value of the Registry key is **C:\Program Files\Microsoft SQL Server\130\DTS\Binn\MsDtsSrvr.ini.xml**. You can update the value of the Registry key to use a different name and location for the configuration file. Note that the version number in the path (120 for SQL Server SQL Server 2014 (12.x), 130 for SQL Server 2016 (13.x), etc.) will vary depending on the SQL Server version.

> **⊗ Caution**
>
> Incorrectly editing the Registry can cause serious problems that may require you to reinstall your operating system. Microsoft cannot guarantee that problems resulting from editing the Registry incorrectly can be resolved. Before editing the Registry, back up any valuable data. For information about how to back up, restore, and edit the Registry, see the Microsoft Knowledge Base article, Description of the Microsoft Windows registry.

The Integration Services service loads the configuration file when the service is started. Any changes to the Registry entry require that the service be restarted.

Connect to the local service

Before you connect to the Integration Services service, the administrator must grant you access to the service.

To connect to the Integration Services Service

1. Open SQL Server Management Studio.
2. Click **Object Explorer** on the **View** menu.
3. On the Object Explorer toolbar, click **Connect**, and then click **Integration Services**.
4. In the **Connect to Server** dialog box, provide a server name. You can use a period (.), (local), or **localhost** to indicate the local server.
5. Click **Connect**.

Connect to a remote SSIS server

Connecting to an instance of Integration Services on a remote server, from SQL Server Management Studio or another management application, requires a specific set of rights on the server for the users of the application.

> **Important**
>
> To connect directly to an instance of the legacy Integration Services Service, you have to use the version of SQL Server Management Studio (SSMS) aligned with the version of SQL Server on which the Integration Services Service is running. For example, to connect to the legacy Integration Services Service running on an instance of SQL Server 2016, you have to use the version of SSMS released for SQL Server 2016. Download SQL Server Management Studio (SSMS).
>
> To manage packages that are stored on a remote server, you do not have to connect to the instance of the Integration Services service on that remote server. Instead, edit the configuration file for the Integration Services service so that SQL Server Management Studio displays the packages that are stored on the remote server.

Connecting to Integration Services on a Remote Server

To connect to Integration Services on a Remote Server

1. Open SQL Server Management Studio.
2. Select **File**, **Connect Object Explorer** to display the **Connect to Server** dialog box.
3. Select **Integration Services** in the **Server type** list.

4. Type the name of a SQL Server Integration Services server in the **Server name** text box.

   > **Note**
   >
   > The Integration Services service is not instance-specific. You connect to the service by using the name of the computer on which the Integration Services service is running.

5. Click **Connect**.

> **Note**
>
> The **Browse for Servers** dialog box does not display remote instances of Integration Services. In addition, the options available on the **Connection Options** tab of the **Connect to Server** dialog box, which is displayed by clicking the **Options** button, are not applicable to Integration Services connections.

Eliminating the "Access Is Denied" Error

When a user without sufficient rights attempts to connect to an instance of Integration Services on a remote server, the server responds with an "Access is denied" error message. You can avoid this error message by ensuring that users have the required DCOM permissions.

To configure rights for remote users on Windows Server 2003 or Windows XP

1. If the user is not a member of the local Administrators group, add the user to the Distributed COM Users group. You can do this in the Computer Management MMC snap-in accessed from the **Administrative Tools** menu.

2. Open Control Panel, double-click **Administrative Tools,** and then double-click **Component Services** to start the Component Services MMC snap-in.

3. Expand the **Component Services** node in the left pane of the console. Expand the **Computers** node, expand **My Computer**, and then click the **DCOM Config** node.

4. Select the **DCOM Config** node, and then select SQL Server Integration Services 11.0 in the list of applications that can be configured.

5. Right-click on SQL Server Integration Services 11.0 and select **Properties**.

6. In the **SQL Server Integration Services 11.0 Properties** dialog box, select the **Security** tab.
7. Under **Launch and Activation Permissions**, select **Customize**, then click **Edit** to open the **Launch Permission** dialog box.
8. In the **Launch Permission** dialog box, add or delete users, and assign the appropriate permissions to the appropriate users and groups. The available permissions are Local Launch, Remote Launch, Local Activation, and Remote Activation. The Launch rights grant or deny permission to start and stop the service; the Activation rights grant or deny permission to connect to the service.
9. Click OK to close the dialog box.
10. Under **Access Permissions**, repeat steps 7 and 8 to assign the appropriate permissions to the appropriate users and groups.
11. Close the MMC snap-in.
12. Restart the Integration Services service.

To configure rights for remote users on Windows 2000 with the latest service packs

1. Run **dcomcnfg.exe** at the command prompt.
2. On the **Applications** page of the **Distributed COM Configuration Properties** dialog box, select SQL Server Integration Services 11.0 and then click **Properties**.
3. Select the **Security** page.
4. Use the two separate dialog boxes to configure **Access Permissions** and **Launch Permissions**. You cannot distinguish between remote and local access - Access permissions include local and remote access, and Launch permissions include local and remote launch.
5. Close the dialog boxes and **dcomcnfg.exe**.
6. Restart the Integration Services service.

Connecting by using a Local Account

If you are working in a local Windows account on a client computer, you can connect to the Integration Services service on a remote computer only if a local account that has the same name and password and the appropriate rights exists on the remote computer.

SSIS Windows service doesn't support delegation

SSIS doesn't support the delegation of credentials, sometimes referred to as a double hop. In this scenario, you're working on a client computer, SSIS is installed on a second computer, and SQL Server is installed on a third computer. Although SSMS successfully passes your credentials from the client computer to the second computer (where SSIS is running), SSIS can't delegate your credentials from the second computer to the third computer (where SQL Server is running).

Configure the firewall

The Windows firewall system helps prevent unauthorized access to computer resources over a network connection. To access Integration Services through this firewall, you have to configure the firewall to enable access.

> **Important**
>
> To manage packages that are stored on a remote server, you do not have to connect to the instance of the Integration Services service on that remote server. Instead, edit the configuration file for the Integration Services service so that SQL Server Management Studio displays the packages that are stored on the remote server.

The Integration Services service uses the DCOM protocol.

There are many firewall systems available. If you are running a firewall other than Windows firewall, see your firewall documentation for information that is specific to the system you are using.

If the firewall supports application-level filtering, you can use the user interface that Windows provides to specify the exceptions that are allowed through the firewall, such as programs and services. Otherwise, you have to configure DCOM to use a limited set of TCP ports. The Microsoft website link previously provided includes information about how to specify the TCP ports to use.

The Integration Services service uses port 135, and the port cannot be changed. You have to open TCP port 135 for access to the service control manager (SCM). SCM performs tasks such as starting and stopping Integration Services services and transmitting control requests to the running service.

The information in the following section is specific to Windows firewall. You can configure the Windows firewall system by running a command at the command prompt, or by setting properties in the Windows firewall dialog box.

For more information about the default Windows firewall settings, and a description of the TCP ports that affect the Database Engine, Analysis Services, Reporting Services, and Integration Services, see Configure the Windows Firewall to Allow SQL Server Access.

Configuring a Windows firewall

You can use the following commands to open TCP port 135, add MsDtsSrvr.exe to the exception list, and specify the scope of unblocking for the firewall.

To configure a Windows firewall using the Command Prompt window

1. Run the following command:

```dos
netsh firewall add portopening protocol=TCP port=135 name="RPC (TCP/135)"
mode=ENABLE scope=SUBNET
```

2. Run the following command:

```dos
netsh firewall add allowedprogram program="%ProgramFiles%\Microsoft SQL
Server\100\DTS\Binn\MsDtsSrvr.exe" name="SSIS Service" scope=SUBNET
```

> **Note**
>
> To open the firewall for all computers, and also for computers on the Internet, replace scope=SUBNET with scope=ALL.

The following procedure describes how to use the Windows user interface to open TCP port 135, add MsDtsSrvr.exe to the exception list, and specify the scope of unblocking for the firewall.

To configure a firewall using the Windows firewall dialog box

1. In the Control Panel, double-click **Windows Firewall**.
2. In the **Windows Firewall** dialog box, click the **Exceptions** tab and then click **Add Program.**
3. In the **Add a Program** dialog box, **click Browse**, navigate to the Program Files\Microsoft SQL Server\100\DTS\Binn folder, click MsDtsSrvr.exe, and then click **Open**. Click **OK** to

close the **Add a Program** dialog box.

4. On the **Exceptions** tab, click **Add Port.**
5. In the **Add a Port** dialog box, type **RPC(TCP/135)** or another descriptive name in the **Name** box, type **135** in the **Port Number** box, and then select **TCP**.

   > **Important**
   >
   > Integration Services service always uses port 135. You cannot specify a different port.

6. In the **Add a Port** dialog box, you can optionally click **Change Scope** to modify the default scope.
7. In the **Change Scope** dialog box, select **My network (subnet only)** or type a custom list, and then click **OK**.
8. To close the **Add a Port** dialog box, click **OK**.
9. To close the **Windows Firewall** dialog box, click **OK**.

   > **Note**
   >
   > To configure the Windows firewall, this procedure uses the **Windows Firewall** item in Control Panel. The **Windows Firewall** item only configures the firewall for the current network location profile. However, you can also configure the Windows firewall by using the **netsh** command line tool or the Microsoft Management Console (MMC) snap-in named Windows firewall with Advanced Security. For more information about these tools, see Configure the Windows Firewall to Allow SQL Server Access.

SQL Server Management Studio Menu Help

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This section contains F1 Help for the dialog boxes and pages available from the menu bar in Microsoft SQL Server Management Studio.

About SQL Server Management Studio

Advanced Save Options

Choose Toolbox Items (Maintenance Tasks Page)

Customize (Commands Page)

Customize (Toolbars Page)

External Tools

Open With (New File)

Save As

Save Changes

Feedback

Was this page helpful? Yes No

Properties Window F1 Help (Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use the **Properties** window to view properties of selected elements. You can also use the **Properties** window to view file, project, and solution properties. The **Properties** window is available by clicking **Properties Window** on the **View** menu.

The **Properties** window displays different types of editing fields, depending on the needs of a particular property. Properties shown in gray are read-only.

Options

Object name
Lists the currently selected object or objects. Only objects from the active editor or designer are visible.

Categorized
Lists all properties and property values for the selected object, by category. You can collapse a category to reduce the number of visible properties. When you expand or collapse a category, you see a plus (+) or minus (-) to the left of the category name. Categories are listed alphabetically.

Alphabetic
Alphabetically sorts all design-time properties and events for selected objects.

Properties
Displays the properties for an object.

Description pane
The description pane appears at the bottom of the **Properties** window and shows the property type and a short description of the property. You can turn the description of the property off and on using the **Description** command on the shortcut menu.

See Also

- Properties Window in SQL Server Management Studio

Feedback

Was this page helpful? Yes No

Options (Environment - Help - General Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use the **Options** dialog box to choose documentation sets and decide whether to display topics internally in Microsoft SQL Server Management Studio or externally in a separate window. To access this dialog box, click **Options** on the **Tools** menu, expand **Help** in the **Environment** node, and then click **General**.

> **Note**
>
> Changes made to the options on this page do not take effect until the environment is restarted.

Settings

Show Help using
Specifies how Help is displayed. Select **External Help Viewer** from the drop-down list to display topics in an external floating window. Select **Integrated Help Viewer** to display Help topics in a Web browser window docked inside SQL Server Management Studio.

Search results per page
Specifies the number of topics that are displayed on a single **Search Results** page.

Show topic abstracts
Specifies whether an abstract for each topic is shown in the **Search Results** list and on the **How do I** tab.

Reuse topic window
When this check box is selected, specifies that Help topics display in the same window so you can navigate among topics using the **Back** and **Forward** buttons. When cleared, this check box specifies that a new window is opened for each topic.

Highlight search terms
Specifies whether search strings are highlighted within **Search Results** topics for local Help.

Online topic Language
Specifies the language with the highest priority in the results list for searches that return results in multiple languages.

Also show English topics
Specifies that English topics are returned in addition to the language selected in the **Topic Language Preference** box.

Feedback

Was this page helpful? Yes No

Options (Environment - General Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use the **Options** dialog box to configure Microsoft SQL Server Management Studio startup actions, general window management options, and other general settings. On the **Tools** menu, click **Options**, expand the **Environment** folder, and then click **General**.

UI element list

Close button affects active tool window only

Specifies that when this check box is selected, only the tool window's focus is closed and not all of the tool windows in the docked set. By default, this check box is selected.

Auto Hide button affects active tool window only

Specifies that when this check box is selected, only the tool window's focus is hidden automatically, not all of the tool windows in the docked set. By default, this check box is cleared.

Feedback

Was this page helpful? Yes No

Options (Environment - Fonts and Colors Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The **Options** dialog box lets you establish a custom font and color scheme for various user interface elements in Microsoft SQL Server Management Studio. On the **Tools** menu, click **Options,** expand the **Environment** folder, and select **Fonts and Colors**.

Color scheme changes do not take effect during the session in which you make them. You can evaluate color changes by opening another instance of SQL Server Management Studio and producing the conditions under which you expect your changes to apply.

UI element list

Show settings for
Lists all the user interface elements for which you can change font and color schemes. After selecting an item from this list you can customize color settings for the item selected in **Display items**.

Expand table

Term	Definition
Text Editor	Changes to font style, size, and color display settings for Text Editor affect the appearance of text in your default text editor. Documents opened in a text editor outside SQL Server Management Studio will not be affected by these settings.
Printer	Changes to font style, size, and color display settings for Printer affect the appearance of text in printed documents. Note: you can select a different default font for printing than that used for display in the Text Editor. This can be useful when printing code that contains both single-byte and double-byte characters.
[All Text Tool Windows]	Changes to font style, size, and color display settings for this item affect the appearance of text in tool windows that have output panes in SQL Server Management Studio. For example, Output window, TextResults window, and so on.

Term	Definition
	Note: Changes to the text of [All Text Tool Windows] items do not take effect during the session in which you make them. You can evaluate such changes by opening another instance of SQL Server Management Studio.
Find Results Window	Changes to font style, size, and color display settings for this item affect the appearance of text in the FindResults window.
Output Window	Changes to font style, size, and color display settings for this item affect the appearance of text in the Output window.
Grid Results	Changes to the font style, size, and color display settings for this item affect the appearance of text in the **Grid Results** area of the Query window.
Execution Plan	Changes to font style, size, and color display settings for this item affect the appearance of text in the Execution Plan of Microsoft SQL Server and SQL Server Compact queries.
Text Results	Changes to font style, size, and color display settings for this item affect the appearance of text in the **Text Results** area of the Query window.
Business Intelligence Designers	Changes to font style, size, and color display settings for this item affect the appearance of text in the BusinessIntelligenceDesignerswindow.

Use Defaults
The **Use Defaults** button resets the value for the default font and color values of the list item that you selected from the **Show settings for** list.

Font (bold type indicates fixed-width fonts)
Lists all the fonts installed on your system. When this drop-down list is first opened, the current font for the element you chose from the **Show settings for** list is selected. Fixed fonts - which are easier to align in an editor - appear in bold.

Size
Lists available point sizes for the selected font. Changing the size of the font affects all **Display items** entries for a **Show settings for** selection.

Display items
Lists the items for which you can modify the foreground and background color.

> ⓘ **Note**
>
> The default display item is Text. As such, properties assigned to Text will be overridden by properties assigned to other display items. For example, if you assign

the color blue to **Text** and the color green to Identifier, all identifiers will appear in green. In this example, Identifier properties override Text properties.

Some display items include:

- Indicator Margin: The margin at the left of the Code Editor where breakpoints and bookmark icons are displayed.
- Collapsible Text: A block of text or code that can be switched into and out of view within the CodeEditor (XML only).

Item foreground
Lists the available colors you can choose for the foreground of the item selected in **Display items**. Because some items are related, a consistent display scheme should be maintained; for example, changing the foreground color of text also changes the foreground color for elements such as String.

Custom
Displays the **Color** dialog box, where you set a custom color for the item selected in the **Display items** list.

> ⓘ **Note**
>
> Your ability to define custom colors may be limited by the color settings for your computer display. For example, if your computer is set to display 256 colors and you select a custom color from the **Color** dialog box, SQL Server Management Studio chooses to the closest available value in **Basic colors** as the default, and displays the color black in the **Color** dialog box.

Item background
Provides a color palette from which you can choose a background color for the item selected in **Display items**. Because some items are related, a consistent display scheme should be maintained; for example, changing the background color of text also changes the background color for elements such as SQL String.

Custom
Displays the **Color** dialog box, where you set a custom color for the item selected in the **Display items** list.

Bold
Select this check box to display the text of selected displayitems in bold font. Bold text is easier to identify in an editor.

Sample

Displays a sample of the font style, size, and color scheme for the selected values in **Show settings for** and **Display items**. You can use this text box to preview the results as you experiment with different formatting options.

See Also

Color Coding in Code Editors

Feedback

Was this page helpful? Yes No

Options (Environment - Keyboard Page)

Article • 02/13/2025

Applies to: SQL Server, Azure SQL Database, Azure SQL Managed Instance, Azure Synapse Analytics, Analytics Platform System (PDW)

The **Options** dialog box lets you customize the keyboard scheme for Microsoft SQL Server Management Studio. Use this option to choose a keyboard scheme. To access this page, click **Options** on the **Tools** menu, select the **Environment** folder, and then click **Keyboard**.

The **Keyboard scheme** drop-down list provides the available collections of shortcut keys. To load a specific scheme, select it from the list. When you are ready to leave the **Options** dialog box, click **OK**. Your selected scheme will be loaded into SQL Server Management Studio.

Standard
The Standard scheme is the default in Microsoft SQL Server and closely resembles the Microsoft Visual Studio shortcut keys.

Use the **Query shortcuts** grid to associate a stored procedure with a keyboard shortcut.

Shortcut
The keyboard combination pressed to execute the stored procedure.

Stored Procedure
The stored procedure that executes when you press the keyboard shortcut.

For a list of the keyboard shortcut keys in each scheme, see SQL Server Management Studio Keyboard Shortcuts.

Feedback

Was this page helpful? Yes No

Options (Environment - Startup page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use the **Options** dialog box to configure Microsoft SQL Server Management Studio startup actions, general window management options, and other general settings. On the **Tools** menu, click **Options**, expand the **Environment** folder, and then click **Startup**.

UI element list

At startup

Select the action for SQL Server Management Studio to perform when it's started. The options are:

- **Open Object Explorer** prompts for a connection and then opens Object Explorer.
- **Open new query window** prompts for a connection and then opens SQL Query Editor.
- **Open Object Explorer and query window** prompts for a connection, then opens both Object Explorer and SQL Query Editor with that connection.
- **Open Object Explorer and Activity Monitor**
- **Open empty environment** opens SQL Server Management Studio without a SQL Query editor window and without connecting Object Explorer to a server.

Hide system objects in Object Explorer

Select this check box to remove the system databases, system tables, system views, and system stored procedures from tree view in Object Explorer. System functions and system data types are not hidden. This option only applies to instances of SQL Server and does not affect servers already connected in Object Explorer.

See also

- Options (Environment - General Page)

Feedback

Was this page helpful? Yes No

Options (Query Execution - General)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

Use this page to specify the options for running Microsoft SQL Server queries. To access this dialog box, right-click the body of a Query Editor window, and then select **Query Options** or on the **Tools** menu, select **Options**, then the **Query Execution** folder.

- **SET ROWCOUNT** The default value of 0 indicates that SQL Server will wait for results until all results are received. Provide a value greater than 0 if you want SQL Server to halt the query after obtaining the specified number of rows. To turn off this option (so that all rows are returned), specify SET ROWCOUNT 0.

- **SET TEXTSIZE** The default value of 2,147,483,647 bytes indicates that SQL Server will provide a complete data field up to the limit of text, ntext, nvarchar(max), and varchar(max) data fields. It doesn't affect the XML data type. Provide a smaller number to limit results of large values. Columns greater than the number provided will be truncated.

- **Execution time-out** Indicates the number of seconds to wait before canceling the query. A value of 0 indicates an infinite wait, or no time-out.

- **Batch separator** Type a word that you use to separate Transact-SQL statements into batches. The default is GO.

- **By default, open new queries in SQLCMD Mode** Select this check box to open new queries in SQLCMD mode. This check box is visible only when the dialog box is opened through the Tools menu.

  When you select this option, be aware of the following limitations:

  - IntelliSense in the Database Engine Query Editor is turned off.

  - Because Query Editor doesn't run from the command line, you can't pass in command-line parameters such as variables.

  - Because Query Editor can't respond to operating-system prompts, you must be careful not to run interactive statements.

- **Reset to Default** Resets all values on this page to the original default values.

- **Prompt to save unsaved T-SQL query windows on close** By default, a prompt appears asking if you want to save changes to file(s) with unsaved changes before closing them or exiting SSMS. Unchecking this option could result in the loss of query editor information when closing query windows or SSMS.
- **Check for open transactions before closing T-SQL query windows** By default, SSMS will check to see if there are open transactions before closing a query editor window. Unchecking this option bypasses this check, potentially resulting in open transactions not being committed.

Feedback

Was this page helpful? Yes No

Options (Designers - Table and Database Designers Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

Use this page to determine the default behavior of the designer. To access the settings, on the **Tools** menu, select **Options**, expand the **Designers** folder, and select **Table Designer**.

> **Note**
>
> The SSMS table designer is not available for tables in Azure Synapse Analytics Dedicated SQL pool (formerly SQL DW).

UI element list

Override connection string time-out value for table designer updates
Permits a new time-out value to be set for the actions of the table designer. This can be useful when the table designer affects a large table and requires extra time to complete the table modification.

Transaction time-out after
Sets a time-out value for the table designer.

Auto generate change scripts
Automatically creates a script to implement the changes defined in the table designer.

Warn on null primary keys
Provides a warning dialog box when a field that is selected for a primary key can contain null values.

Warn about difference detection
If you check this box, when you save the changes, a message box will list any conflicts between your changes and changes that were made by another user.

Warn about tables affected
Provides a warning dialog box that lists the tables to be affected by the action and prompts for confirmation.

Prevent saving changes that require table re-creation
Prevents a user from making changes that require re-creating the table. The following actions might require a table to be re-created:

- Adding a new column to the middle of the table
- Dropping a column
- Changing column nullability
- Changing the order of the columns
- Changing the data type of a column

Default table view
Select the way you want to see tables in the designers:

- **Standard**

  Shows the table header, all column names, data types, and the Allow Nulls setting.

- **Column Names**

  Shows the column names.

- **Key**

  Shows the table header and the primary key columns.

- **Name Only**

  Shows only the table header with its name.

- **Custom**

  Allows you to choose which columns to view.

Launch add table dialog on new diagram
Automatically opens the **Add Table** dialog box when the designers open.

Feedback

Was this page helpful? Yes No

Options (Designers - Analysis Services Designers - General)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use the **Designers**, **Maintenance Plans**, **Analysis Services**, **General** page to determine the default behavior of the Analysis Services Designers.

Connectivity

Query Timeout
The number of seconds that the designer waits for the query to respond before generating an error.

Connection Timeout
The number of seconds that the designer waits for a connection to be confirmed before generating an error.

Feedback

Was this page helpful? Yes No

Options (Azure Services)

Article • 02/13/2025

Use this page to specify options related to Azure cloud services. To access this dialog box, go to **Tools** > **Options** > **Azure Services** from the top menu bar.

> **Note**
>
> While Microsoft Entra ID is the new name for Azure Active Directory (Azure AD), to prevent disrupting existing environments, Azure AD still remains in some hardcoded elements such as UI fields, connection providers, error codes, and cmdlets. In this article, the two names are interchangeable.

Miscellaneous

Expand table

Option	Information	Description
Allow single sign-on to SQL using Microsoft Entra authentication	**True** **False**	When true, removes the requirement to provide full credentials to connect to Azure SQL Database and SQL Server using Microsoft Entra ID if using the same account used to log in to Windows.
Azure Data Factory Portal URL	`https://adf.azure.com/`	Specifies the URL for the Azure Data Factory portal.
Gallery Endpoint	`https://gallery.azure.com/`	Specifies the endpoint for the Resource Manager gallery of deployment templates.
Graph Audience	`https://graph.microsoft.com/`	Graph Endpoint Resource ID.
Graph Endpoint	`https://graph.microsoft.com/`	Specifies the URL for Azure Active Directory Graph requests.
Management Portal URL	`https://portal.azure.com`	Specifies the URL for the Management Portal.

Option	Information	Description
MSAL Output Window Trace Level	**Information** **None** **Verbose** **Warning**	The level of Microsoft Authentication Library (MSAL) login tracing to send to the Output window.
Publish Setting File URL	`https://go.microsoft.com/fwlink/?LinkID=335839`	Specifies the URL from which the `.publishsettings` file can be downloaded.
SQL Database Service Principal Name	`https://database.windows.net/`	The Azure SQL Database SPN to obtain a token when using Azure AD authentication. Also the audience of the JSON Web Token (JWT) for server-side JSON Web Token (JWT) parsing/validation.
Use system default web browser	**True** **False**	When true, SSMS will launch the user's default web browser to perform Azure logins. When false, SSMS will use an embedded browser. This option defaults to True with new installs starting with SSMS 19.1
Use Web Account Manager	**True** **False**	Enable this option if your enterprise uses FIDO keys or other advanced options not supported by a web browser. For example, a company might implement a conditional access policy, which prevents authentication to Azure using a browser. If you aren't sure of the appropriate value for your environment, contact your SQL Database Administrator or your IT team.

Resource Management

Expand table

Option	Information	Description
Microsoft Entra Authority	`https://login.microsoftonline.com/`	Specifies the base authority for Microsoft Entra authentication.

Option	Information	Description
Microsoft Entra Endpoint Resource ID	`https://management.core.windows.net/`	Specifies the audience for tokens that authenticate requests to Azure Resource Manager (ARM) or Service Management (RDFE) endpoints.
Resource Manager Endpoint	`https://management.azure.com/`	Specifies the URL for Resource Management requests.
Service Endpoint	`https://management.core.windows.net/`	Specifies the endpoint for Service Management requests.

Select an Azure Cloud

Select the Azure Cloud to which Management Studio connects to discover and manage resources.

Expand table

Option	Description
Azure China Cloud	International public cloud service generally available in the China market. For more information, see Azure China documentation.
Azure Cloud	Default public cloud.
Azure US Government	Dedicated cloud for US government agencies and their partners. For more information, see Azure Government documentation.
Custom	Select **Custom** to enter custom URLs and DNS suffixes. The options listed on the page become editable.

Service Addresses

Expand table

Option	Information	Description
Azure Data Lake Services Endpoint	`azuredatalakeanalytics.net`	Specifies the Azure Data Lake Analytics catalog and job endpoint suffix.
Azure Data Lake Store File System Endpoint	`azuredatalakestore.net`	Specifies the Azure Data Lake Store files system endpoint suffix.

Option	Information	Description
Azure Key Vault Audience	`https://vault.azure.net/`	Specifies the audience for access tokens that authorize requests for Key Vault services.
Azure Key Vault DNS Suffix	`vault.azure.net`	Specifies the domain name suffix for Azure Key Vault servers.
Azure Key Vault Managed HSM Audience	`https://managedhsm.azure.net/`	Specifies the audience for access token that authorize requests for Key Vault Managed HSM (Hardware Security Module) services.
SQL Database DNS Suffix	`.database.windows.net`	Specifies the domain name suffix for Azure SQL Database servers.
Storage Endpoint	`core.windows.net`	Specifies the endpoint for storage access. The option includes, blobs, tables, queues, and file storages.
Traffic Manager DNS Suffix	`trafficmanager.net`	Specifies the domain name suffix for Azure Traffic Manager services.

Feedback

Was this page helpful? Yes No

Options (SQL Server Object Explorer - Commands)

Article • 02/13/2025

Use this page to specify options related to commands available from Object Explorer. To access this dialog box, go to **Tools** > **Options** > **SQL Server Object Explorer** > **Commands** from the top menu bar.

Audit Log Viewer options

Expand table

Option	Information	Description
Value for Select Top <n> Audit records command	1000	Specifies the number of returned rows from a server audit on the log viewer. Specifying a value of zero (0) returns all rows (not recommended).

Azure Data Studio options

Expand table

Option	Information	Description
Full path the Azure Data Studio instance to invoke		When specified, SSMS uses it to launch a new instance of Azure Data Studio (for example, C:\Program Files\Azure Data Studio\bin\azuredatastudio.com). Leave blank to let SSMS use its heuristic.

Connection security

Expand table

Option	Information	Description
Trust server certificate for imported connections	**Always** **Prompt** **Never**	When set to **Always**, existing connections in the Most Recently Used (MRU) list that were imported after upgrading from SSMS 19.x or earlier will have **Trust Server Certificate** enabled. The default value is **Prompt**.

Drag/drop

Expand table

Option	Information	Description
Prepend dragged object name with schema and period	**True** **False**	Disable to not include the schema name when dragging objects from Object Explorer.
Surround object names with brackets when dragged	**True** **False**	Disable to not surround object names with brackets when dragging objects from Object Explorer. Object names with spaces or closing square brackets are always escaped, regardless of this setting.

PowerShell options

Expand table

Option	Information	Description
Additional parameters to use when importing the SQL Server module		When specified, PowerShell uses these parameters when running `Import-Module SQLServer`. Parameters can be used to force the import of a specific version of the module (for example, `-RequiredVersion 22.0.59`)
Execution policy to use	**AllSigned** **Bypass** **RemoteSigned** **Restricted** **Undefined** **Unrestricted**	When specified, this value is passed as an argument to `-ExecutionPolicy` when a new PowerShell session is started. Leave blank to use RemoteExecution.
Path to the PowerShell instance to invoke		When specified, SSMS uses it to launch a new instance of PowerShell (for example, `C:\Program Files\PowerShell\7\pwsh.exe`). The path can be an absolute or relative path. Use this parameter to force using PowerShell 7. Leave blank to use the 64-bit PowerShell 5.
Skip check on minimum required version of the module	**True** **False**	When true, SSMS skips the validation of the minimum required version of the `SQLServer` module. We recommend that you leave this value as `False`.

Rename options

Expand table

Option	Information	Description
Prompt to confirm the renaming of schema objects	**True** **False**	When true, an attempt to rename a database object using Object Explorer prompts for confirmation unless using the context menu.

Table and View options

Expand table

Option	Information	Description
Value for Edit Top <n> Rows command	200	Specifies the number of returned rows using the TOP clause for the Edit command. Specifying a value of zero (0) returns all rows (not recommended).
Value for Select Top <n> Rows command	1000	Specifies the number of returned rows using the TOP clause for the Select command. Specifying a value of zero (0) returns all rows (not recommended).

Task dialogs

Expand table

Option	Information	Description
Remember task dialog position	**True** **False**	Enable this option to remember the position of a task dialog or property sheet when it closes. When the dialog or sheet reopens, it restores to the remembered position.

Feedback

Was this page helpful? Yes No

Options (Output Window - General)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

Use this page to set the default behavior of the output window.

> **Note**
>
> While Microsoft Entra ID is the new name for Azure Active Directory (Azure AD), to prevent disrupting existing environments, Azure AD still remains in some hardcoded elements such as UI fields, connection providers, error codes, and cmdlets. In this article, the two names are interchangeable.

The Output Window can be opened from the View menu. To access the settings, on the **Tools** menu, select **Options**, and expand the **Output Window** folder or use the key combination Ctrl+Alt+O.

There are multiple logging channels available to provide more information when using SQL Server Management Studio.

The following table gives an overview of the types of messages associated with each output channel.

Expand table

Channel	Definition
Object Explorer	This channel outputs the query text and elapsed times of SQL queries needed to expand nodes in Object Explorer. Each query logs a Begin Query and an End Query event. Each event has a timestamp and the URN associated with the entity being queried. The URN refers to the underlying SQL Management Object and consists of an XPath-style hierarchy. For example, the URN for a table named "Table1" in database "Db" on server "MyServer" would be "Server[@Name='MyServer']/Database[@Name='Db']/Table[/@Name='Table1']". Expanding one node in Object Explorer can require multiple such queries with different parameters. The End Query event contains the elapsed time of the query along with the T-SQL text. This query data is useful for server performance analysis in cases where Object Explorer seems unusually slow to expand a particular node. Note: only some nodes in Object Explorer provide this query detail when expanding. Enabled by default.

Channel	Definition
Telemetry	Telemetry is the stream of anonymous feature usage data collected by Microsoft. These events could be useful for your tracking of SSMS usage. It can help you identify what Object Explorer nodes you expand and what commands run during your SSMS session while the Output window is open. Enabled by default.
SQL Client	This channel displays information from the Microsoft.Data.SqlClient data provider. This option requires a restart of SSMS for output to appear.
Azure Active Directory Interactive authentication	For cases where Microsoft Entra authentication is used, this channel outputs information specific to the authentication process.
Azure REST API	This channel starts after authentication with Microsoft Entra credentials occurs. This channel outputs information specific to Azure REST API calls when Azure-connected features are used and more troubleshooting information is needed.
Activity Monitor	This channel starts when Activity Monitor opens for a server. This stream contains events showing part of the query text and timestamp of each query, error messages, and notifications of the monitor being paused due to connectivity problems. If Activity Monitor is idle or otherwise failing to update, check this output channel for more information.

Feedback

Was this page helpful? Yes No

Options (SQL Server Object Explorer - Scripting)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use this page to set scripting options that apply to the following commands on object context menus in **Object Explorer**:

- **Edit** commands for user tables and views.
- **Script <object> as** commands for user-created objects.
- **Modify** command for user-created objects.
- This page also sets the scripting option defaults for the **Generate SQL Server Script Wizard**.

Remarks

The **Edit** and **Modify** commands might produce results that are different from the **Script <object> as** command for the same option setting. The **Edit** and **Modify** commands are designed to modify objects in the current database during a Query Editor session. The **Script <object> as** command is designed to generate a script so that it can be used later to create objects.

Options

Specify scripting options by selecting from the available settings in the list to the right of each option.

> **Note**
>
> The default settings listed only apply to the **Script entire database and all database objects** option and may vary when using the **Select specific database objects** option.

General scripting options

Delimit individual statements
Separates individual Transact-SQL statements by using a batch separator. To change the default batch separator for **Query Editor**, select **Tools/Options/Query Execution/SQL Server/General/Batch separator**. Default is False. For more information, see GO (Transact-SQL).

Include descriptive headers
Adds descriptive comments to the script by separating the script into sections for each object. Default is True. For more information, see /.../ (Comment) (Transact-SQL).

Include enabling vardecimal compression
Includes the vardecimal storage options. Default is False. For more information, see sp_db_vardecimal_storage_format (Transact-SQL).

Script change tracking
Includes change tracking information in the script.

Script full-text catalogs
Includes a script for full-text catalogs. Default is False. For more information, see CREATE FULLTEXT CATALOG (Transact-SQL).

Script USE <database>
Adds the USE DATABASE statement to the script to create database objects in the context of the current **Object Explorer** database. When the script is expected for use in a different database, select False to omit. Default is True. For more information, see USE (Transact-SQL).

Object scripting options

Check for object existence Check that an object with the given name exists before dropping or altering or that an object with the given name doesn't exist before creating. For more information, see IF...ELSE (Transact-SQL) and EXISTS (Transact-SQL).

Generate script for dependent objects
Generates a script for other objects that are required when the script for the selected object is executed. Default is False.

Schema qualify object names
Qualifies object names with the object schema. Default is False. For more information, see Create a Database Schema.

Script data compression options Includes data compression options in the script. Default is False.

Script extended properties
Includes extended properties in the script if the object has extended properties. Default is False. For more information, see sp_addextendedproperty (Transact-SQL).

Script owner
Includes the owner in the generated script. Default is False.

Script permissions
Includes permissions on database objects in the script. Default is True. For more information, see Permissions.

Table/View options

The following options apply only to scripts for tables or views.

Convert user-defined data types to base types
Converts user-defined data types to the base types from which they were created. Use True when the source database user-defined data types don't exist in the database where the script is run. Use False to keep the user-defined data types. Default is False. For more information, see CREATE TYPE (Transact-SQL).

Generate SET ANSI PADDING commands
Adds the SET ANSI_PADDING statement before and after each CREATE TABLE statement. Default is True. For more information, see SET ANSI_PADDING (Transact-SQL).

Include collation
Includes collation in column definition. Default is True. For more information, see Collation and Unicode Support.

Include IDENTITY property
Includes definitions for IDENTITY seed and IDENTITY increment. Default is True. For more information, see IDENTITY (Property) (Transact-SQL).

Schema qualify foreign key references
Adds the schema name to table references for FOREIGN KEY constraints. Default is True.

Script bound defaults and rules
Includes the **sp_bindefault** and **sp_bindrule** binding stored procedure calls. Default is True. For more information, see sp_bindefault (Transact-SQL) and sp_bindrule (Transact-SQL).

Script CHECK constraints
Adds CHECK constraints to the script. Default is True.

Script defaults
Includes column default values in the script. Default is False. For more information, see CREATE DEFAULT (Transact-SQL).

Script file groups
Specifies the filegroup in the ON clause for table definitions. Default is False. For more information, see CREATE TABLE (Transact-SQL).

Script foreign keys
Includes FOREIGN KEY constraints in the script. Default is False.

Script full-text indexes
Includes full-text indexes in the script. Default is False. For more information, see CREATE FULLTEXT INDEX (Transact-SQL).

Script indexes
Includes clustered, nonclustered, and XML indexes in the script. Default is True. For more information, see CREATE INDEX (Transact-SQL).

Script partition schemes
Includes table partitioning schemes in the script. Default is False. For more information, see CREATE PARTITION SCHEME (Transact-SQL).

Script primary keys
Includes Primary and Foreign Key Constraints in the script. Default is True.

Script statistics
Includes user-defined statistics in the script. Default is False. For more information, see CREATE STATISTICS (Transact-SQL).

Script triggers
Include triggers in the script. Default is False. For more information, see CREATE TRIGGER (Transact-SQL).

Script unique keys
Includes Unique Constraints and Check Constraints in the script. Default is False.

Script view columns
Declares view columns in view headers. Default is False. For more information, see CREATE VIEW (Transact-SQL).

Include dri system names
Includes system generated constraint names to enforce declarative referential integrity. Default is False. For more information, see REFERENTIAL_CONSTRAINTS (Transact-SQL).

Version options

Match script settings to source If enabled the target version, engine edition and engine type of the scripts generated are set to the values of the server the object being scripted. This configuration disables (and ignores) the other version options.

Script for database engine edition Scripts generated are targeted for the specified Engine Edition.

Script for database engine type Scripts generated are targeted for the specified Database Engine Type.

Script for server version
Scripts generated are targeted for the specified version of SQL Server. Features that are new in SQL Server can't be scripted for earlier versions. Some scripts that are created for SQL Server can't be executed on servers that are running on an earlier version of SQL Server, or on a database that has an earlier database compatibility level setting.

See also

Generate Scripts (SQL Server Management Studio)

Feedback

Was this page helpful? Yes No

External Tools

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use this dialog box to add external tools, such as SQL Server Configuration Manager or Notepad, to the **Tools** menu. Adding external tools allows you to easily launch other applications while working in SQL Server Management Studio. You can specify arguments and a working directory when launching the tool. In addition, the outputs from some tools can be displayed in the Output window. The **External Tools** dialog box is available on the **Tools** menu.

Options

Menu Contents
Lists the titles of the items currently added to the **Tools** menu. Use the **Move Up** and **Move Down** arrows to change the order the items that appear on the menu. Use the **Delete** button to remove an item from the menu.

Move Up
Move the selected tool higher in the list of tools that appear on the **Tools** menu.

Move Down
Move the selected tool lower in the list of tools that appear on the **Tools** menu.

Add
Clear the text boxes so you can specify a new tool.

Delete
Remove the tool or command from the **Menu Contents** list as well as from the **Tools** menu.

Title
The name of the tool or command that will appear on the **External Tools** submenu of the **Tools** menu. Place an ampersand before a letter in the name of the tool to use that letter as an accelerator key for the tool. For example, `&Spy++` would display **Spy++** on the **Tools** menu.

Command
Specify the path to the .exe, .com, .pif, .bat, .cmd, or other file that you intend to launch.

The output from `.bat`, `.com`, and other files can be viewed in the Output window if you select the **Use output window** check box.

Arguments
Specify the variables that are passed to the tool when the tool is selected on the menu. Arguments can specify values that are passed to the tool or command when it is launched. For example, a value can specify a file name or directory. Use the **Arrow** button to select from a list of predefined arguments. You can add more than one. For a complete list of predefined arguments and their definitions, see Arguments for External Tools. You can also enter custom arguments (command-prompt switches, for example), depending on the command or tool you use.

Initial directory
Specify the working directory of the tool. Use the **Arrow** button to select directories. You can select more than one.

Use output Window
Specify whether or not results from the tool are displayed in the Output window. This option is only available for files, such as .bat and .com files, that normally display output in the command prompt window. When enabled, this option eases window management when you are following the progress of a tool.

Prompt for arguments
Display the **Arguments** dialog box to enable you to enter or edit values for the arguments each time you launch the external tool.

Treat output as Unicode
Allow the Output window to accept Unicode.

Close On Exit
Close the window opened by the tool when the tool is closed.

Example

To add SQL Server Configuration Manager to the Tools menu

1. On the **Tools** menu, click **External Tools**.
2. In the **Title** box, type **SQL Server Configuration Manager**.
3. In the **Command** box, type the path to the Microsoft Management Console executable, such as **C:\WINNT\system32\mmc.exe**

4. In the **Arguments** box, type the path to the .msc file, such as **"C:\WINNT\system32\SQLServerManager.msc"**

> **Note**
>
> View the properties of the SQL Server Management Studio shortcut on the **Start** menu to confirm the location of the files on your computer.

Feedback

Was this page helpful? Yes No

Customize (Commands Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This dialog box enables you to add and remove commands on toolbars and menus as well as change the images used for toolbar buttons or menu commands. You can access the **Commands** page by clicking **Customize** on the **Tools** menu and then clicking **Commands**.

UI element list

Categories
Specifies the set of commands that are displayed in the **Commands** list box. The categories of commands are based on menu titles provided by the tools and designers that the environment is currently supporting. This list of titles is dynamic so that the order of categories and the menu titles change, depending on the tools and the designer, as well as any customizations made to them. Therefore, it is possible for two menus from different designers to have the same name, so the same title can appear twice but offer different command sets.

Commands
Displays the commands and command images based on the category you selected. You can drag a command onto the toolbar you want to customize.

Modify Selection
Displays a list of commands you can use to customize the display of the button on the toolbar. For example, you can change the image or accelerator keys, as well as specify whether to display text instead of an image for the command. This button is available after you select a command button on a toolbar you want to customize.

Rearrange Commands
Displays the **Rearrange Commands** dialog box, which allows you to rename, reorder, add, and remove commands on menus and toolbars as well as change button and command images.

Keyboard
Displays the **Keyboard** page of the **Options** dialog box so you can specify shortcut key combinations for commands.

See Also

Customize Menus and Shortcut Keys

Feedback

Was this page helpful? Yes No

Choose Toolbox Items (Maintenance Tasks Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This tab of the **Customize Toolbox** dialog box displays a list of all maintenancetask components registered on your computer and makes it possible for you to change the ones that are displayed in the Toolbox. You can open the **Customize Toolbox** dialog box from the **Tools** menu. To sort the list of components, select any column heading.

The **Maintenance Tasks** tab includes the following columns of information.

Name
Displays the names of available components. Preceding each name is a check box. If selected, a check box indicates that SQL Server Management Studio has found an entry for the component in your computer's registry. The component is either already displayed on the active **Toolbox** tab, or it will be added to it when you click **OK**. If cleared, a check box indicates that the component is not currently displayed in the **Toolbox**, or that it will be removed from the **Toolbox** when you click **OK**.

Path
Displays the full path to the component. To identify the default components that shipped with the product, sort on this column and then locate those stored on the Microsoft Visual Studio installation path.

Last Modified
Displays the date when the component was last modified.

Click on a name to show the attributes of the component in the **Language** and **Version** boxes, along with the icon.

Language
The language of the component.

Version
The version of the component.

Feedback

Was this page helpful? Yes No

About SQL Server Management Studio

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics

Accessed from the Help menu in SQL Server Management Studio (SSMS), you can use this dialog box to view or copy information about your installation.

Grid Layout

Component Name
Lists the name of the installed components:

- **SQL Server Management Studio**
- **SQL Server Management Objects (SMO)**
- **Microsoft T-SQL Parser**
- **Microsoft Analysis Services Client Tools**
- **Microsoft Data SqlClient (MDS)**
- **Microsoft SQL Server Data-Tier Application Framework (DacFX)**
- **Microsoft .NET Framework**
- **Operating System**

Versions
Lists the version of the installed component.

Copy Info
Copies the information in the grid to the clipboard.

Feedback

Was this page helpful? Yes No

Save Changes

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Warns you that changes to the items listed will not be saved unless you click **Yes**.

This dialog box can appear when you are closing solutions, maintenance plans, or other open files.

To display the Save Changes dialog box

- With a new file open, on the **File** menu, click **Close**.

Changes to previously saved items will be saved automatically to their save location in storage. If an item has not yet been saved, the **Save File As** dialog box appears to prompt you for a new save location.

Options

Save changes to the following items?
Selected items will be saved. Items can be selected and cleared.

> **Note**
>
> When saving, specify only the object name, not the schema and object name.

Feedback

Was this page helpful? Yes No

Open With (New File)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can open a document in one or more editors by clicking **Open** on the **File** menu and then clicking **File**. In the **Open File** dialog box, select the file, click the **Open** arrow, and then click **Open With**. In the **Open With** dialog box, in the **Select a program to open** list, click the preferred program, and then click **Open**.

Options

Select a program to open
Lists the editors available in Microsoft Visual Studio for the selected type of file. Choose the editor with which to open the document from the list displayed, or click **Add** to include a new editor in the list.

Open
Click **Open** to open the document in the selected editor.

Add
Click this button to add a program to the list under **Select a program to open**. You can either type the file path to the program in the **Program Name** field or browse to the program's location by clicking **Browse**. In **Friendly Name**, enter a program name to display in the list under **Select a program to open**.

Remove
To remove a program, select the program and then click **Remove**.

Set as Default
To specify a default editor (and language encoding options, if applicable) for the type of file selected, choose a program from the list under **Select a program to open** and then click **Set as Default**. The next time you open this type of file in SQL Server Management Studio, the document will open in the new default editor.

> **Note**
>
> In the list of programs under **Select a program to open**, the name of the default editor for the type of file selected is followed by **(Default)**.

See Also

Associating File Extensions to a Code Editor

Feedback

Was this page helpful? Yes No

Advanced Save Options

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use this dialog box to select a language encoding scheme and to specify the characters to be inserted at the end of each line. To access the **Advanced Save Options** dialog box, on the **File** menu, click **Save** <query name>**.sql As** to open the **Save File As** dialog box, click on the down arrow on the **Save** button, and then click **Save with Encoding**.

Options

Encoding

Use this list box to apply a new language encoding scheme to the target file.

Line Endings

To ensure that the target file will run on a specific operating system, select whether a carriage return (for Macintosh systems), a line feed (for Unix systems), or both characters (for Microsoft Windows systems) will mark the end of each line.

Feedback

Was this page helpful? Yes No

Save As

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Used to save an instance of the current item at a specified location in a specified file format. Click **Save** <*file*> **As** on the **File** menu (where <*file*> is the name of the current item), or press ALT+F, A in the Code Editor.

Central Panel

Save in
Locate the existing project folder from this drop-down menu. Selecting a folder from this list displays the contents of the folder in the primary pane below.

File name
Use this option to view the current file name, change the file name, or filter the files and folders that are displayed. To filter the files and folders that are displayed, enter a full or partial file name on which to filter. You can use the asterisk (`*`) as a wildcard.

> **Tip**
>
> To display files on Web and network locations, enter a URL or network path in the **File name** box. For example, `https://mywebsite` displays the files available at the "mywebsite" Web location and "\\myserver\myshare" displays the files available at the "myshare" location on "myserver".

Save as type
Use this option to select a new file type for the selected item. The file types displayed include all available file types to which the selected item can be converted.

Advanced Save Options
To access the **Advanced Save Options Dialog Box**, select the small rectangle at the right of the **Save** button and then click **Save with Encoding**. Use this dialog box to specify an encoding for the file and the characters to use at Line endings.

Left Panel

Desktop
Displays any files and folders located on the desktop.

My Projects
Displays files and folders at the **My Projects** or the most recently visited location.

My Computer
Displays the **My Computer** location on your computer.

See Also

Advanced Save Options
Manage Files with Encoding

Feedback

Was this page helpful? Yes No

Customize (Toolbars Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The **Toolbars** tab enables you to create, rename, remove, and reset toolbars. You can access this tab by clicking **Customize** on the **Tools** menu and then clicking **Toolbars**.

UI element list

Toolbars
Displays the built-in toolbars and any other toolbars you create. When a toolbar is displayed, a check mark appears to the left of it in this dialog box.

> **Note**
>
> The menu bar cannot be hidden. It can only be reset.

New
Displays the **New Toolbar** dialog box, which allows you to create and name a custom toolbar.

Rename
Displays the **Rename Toolbar** dialog box, which allows you to change the name of a custom toolbar only.

Delete
Deletes the custom toolbar selected in the **Toolbars** list.

Reset
Removes any changes to the predefined toolbar selected in the **Toolbars** list and resets it to its original state. It is available only if you select a built-in toolbar.

Large icons
Increases the size of toolbar icons so that they are easier to see.

Show ScreenTips on toolbars
Displays on-screen descriptions of toolbar buttons when the pointer hovers over them.

Show shortcut keys in ScreenTips
Specifies whether shortcut key combinations appear in ToolTips for commands on

toolbars.

Keyboard
Displays the **Keyboard** pane of the **Options** dialog box, which allows you to change and define shortcut key combinations for commands.

Feedback

Was this page helpful? Yes No

Options (Designers - Maintenance Plans Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use the **Options** dialog box to determine the default behavior of the designer when new shapes are added to the design surface. On the **Tools** menu, click **Options**, expand **Designers**, and select **Maintenance Plans**.

UI element list

Connect a new shape to the selected shape by default
When this check box is selected, new shapes automatically connect to other shapes when they are added to the design surface. When this check box is cleared, new shapes are not connected when they are added. Selecting this check box enables the following boxes.

Specify the connection of the new shape
Select from this drop-down list box to designate the default for the type of connection made between shapes. The three choices are:

- **Use a Success constraint for new shape**
- **Use a Failure constraint for new shape**
- **Use a Completion constraint for new shape**

Specify the location of the new shape
Select from this drop-down list box to designate the default location where a new shape is added to the design surface. The four choices are:

- **Add the new shape to the left of the selected shape**
- **Add the new shape to the right of the selected shape**
- **Add the new shape above the selected shape**
- **Add the new shape below the selected shape**

Feedback

Was this page helpful? Yes No

Options (Environment - Help - Online Page)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The **Options** dialog box lets you specify the Help sources to use for Microsoft SQL Server Management Studio searches, including F1 Help. On the **Tools** menu, click **Options**, expand the **Environment** folder, select **Help**, and then click **Online**. If the **Online** page does not appear in the list, select the **Show all settings** check box in the **Options** dialog box.

When loading Help content
Sets the order in which Help sources are used for F1 Help, and specifies which Help sources are used.

- **Try online first, then local**
- Help from the online sources are located first, then help from the local computer.
- **Try local first, then online**
- Help from the local computer is located first, then online sources.
- **Try local only, not online**

  Only the Help installed on your computer is used for F1 Help.

Search these providers
Specifies the order in which Help sources are used when searching Help by means of keywords.

Codezone Community
Lists the possible Codezone Community Web sites that can be included in Help searches.

Details
Displays a brief description for each provider selected in the **Codezone Community** list if available.

Read the privacy statement
Displays the Microsoft privacy policy for online use.

Feedback

Was this page helpful? Yes No

Object Explorer

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

SQL Server Management Studio provides features for managing objects in instances of the Database Engine, Analysis Services, Integration Services, and Reporting Services.

Benefits of Object Explorer

Object Explorer provides a hierarchical user interface to view and manage the objects in each instance of SQL Server. The Object Explorer Details pane presents a tabular view of instance objects, and the capability to search for specific objects. The capabilities of Object Explorer vary slightly depending on the type of server, but generally include the development features for databases, and management features for all server types.

Object Explorer Tasks

Expand table

Description	Topic
Describes how to open the Object Explorer and configure the options that define the behavior of the explorer.	Open and Configure Object Explorer
Describes how to connect Object Explorer to an instance of the Database Engine, Analysis Services, Reporting Services, and Integration Services.	Connect to an Instance From Object Explorer
Describes how to manage objects represented as nodes in the Object Explorer hierarchy.	Manage Objects by Using Object Explorer
Describes the Object Explorer Details Pane, a tabular view of all of the objects in the server with a user interface to manage them.	Object Explorer Details Pane
Describes ways to run custom reports in SQL Server Management Studio.	Custom Reports in Management Studio

Feedback

Was this page helpful? Yes No

Add Steps to a SQL Server Agent Master Job

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes how to add steps to a SQL Server Agent master job in SQL Server by using SQL Server Management Studio or Transact-SQL.

In This Topic

- **Before you begin:**

  Limitations and Restrictions

  Security

- **To add steps to a SQL Server Agent master job, using:**

  SQL Server Management Studio

  Transact-SQL

Before You Begin

Limitations and Restrictions

A SQL Server Agent master job cannot be targeted at both local and remote servers.

Security

Permissions

Unless you are a member of the **sysadmin** fixed server role, you can only modify jobs that you own. For detailed information, see Implement SQL Server Agent Security.

Using SQL Server Management Studio

To add steps to a SQL Server Agent master job

1. In **Object Explorer,** click the plus sign to expand the server that contains the job to which you want to add steps.
2. Click the plus sign to expand **SQL Server Agent**.
3. Click the plus sign to expand the **Jobs** folder.
4. Right-click the job to which you want to add steps and select **Properties**.
5. In the **Job Properties** *-job_name* dialog box, under **Select a page**, select **Steps**. For more information on the available options on this page, see Job Properties - New Job (Steps Page).
6. When finished, click **OK**.

Using Transact-SQL

To add steps to a SQL Server Agent master job

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. Copy and paste the following example into the query window and click **Execute**.

```
-- creates a job step that changes database access to read-only for the
Sales database.
-- specifies 5 retry attempts, with each retry to occur after a 5
minute wait.
-- assumes that the Weekly Sales Data Backup job already exists
USE msdb;
GO
EXEC sp_add_jobstep
    @job_name = N'Weekly Sales Data Backup',
    @step_name = N'Set database to read only',
    @subsystem = N'TSQL',
    @command = N'ALTER DATABASE SALES SET READ_ONLY',
    @retry_attempts = 5,
    @retry_interval = 5 ;
GO
```

For more information, see sp_add_jobstep (Transact-SQL).

Feedback

Was this page helpful? Yes No

Manage objects by using Object Explorer

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can use Object Explorer in SQL Server Management Studio (SSMS) to manage objects such as databases, tables, and stored procedures.

Viewing objects in Object Explorer

Object Explorer uses a tree structure to group information into folders. To expand folders, Select the plus sign (+) or double-click the folder. Expand folders to show more detailed information. Right-click folders or objects to perform common tasks. Double-click objects to perform the most common task.

The first time you expand a folder, Object Explorer queries the server for information to populate the tree. You can perform other functions while the tree is populating. While Object Explorer is populating the tree, you can select **Stop** to halt the process. Further actions, such as filtering the list, will only act upon the portion of the folder that was populated, unless you refresh the folder to start population again.

To conserve resources when there are many objects, the folders in the Object Explorer tree don't automatically refresh their list of contents. To refresh the list of objects within a folder, right-click the folder, and then select **Refresh**.

Object Explorer can only display up to 65,536 objects. After you have exceeded 65,536 visible objects, you can't scroll through additional objects in the Object Explorer tree view. To view additional objects in Object Explorer, close nodes that you aren't using or apply filtering to reduce the number of objects.

Filtering the list of objects in Object Explorer

When a folder contains a large number of objects, it might be difficult to find the object you're looking for. In such cases, use the filter feature of Object Explorer to reduce the list to a smaller size. For example, you might want to find a specific database user or the most recently created table in lists that contain hundreds of objects. Select the folder that you want to filter, and then select the filter button to open the **Filter Settings** dialog

box. You can filter the list by name, create date, and sometimes schema, and provide extra filtering operators like **Starts with**, **Contains**, and **Between**.

Multi-select

Only one object can be selected at a time in Object Explorer. To select multiple items, press **F7** to open the **Object Explorer Details Page**. The **Object Explorer Details Page** supports multi-select.

Register a server from Object Explorer

When connected to a server, you can easily register the server for future use. In Object Explorer, right-click the server name, and then select **Register**. In the **Register Server** dialog box, specify where in the server group tree you want to place the server. In the **Server name** box, you can replace the server name with a more meaningful server name. For example, you could register server **APSQL02** with a more meaningful name such as "**Accounts Payable**".

Performing actions on Object Explorer nodes

You perform actions on objects by right-clicking the Object Explorer node representing the object. Each type of object supports a unique set of right-click actions. Some of the types of actions you can perform by using the right-click menus include:

Open a connected Query Editor

When Object Explorer is connected to a server, you can open a new Code Editor window using the connection settings of Object Explorer. To open a new Code Editor window, right-click the server name in Object Explorer, and then select **New Query**. To open a Code Editor window using a particular database, right-click the database name, and then select **New Query**. When opening a new query for an Analysis Services server, you can select DMX, MDX, or XMLA queries.

Start PowerShell

You can start a PowerShell session by right-clicking most folders and objects in the Object Explorer tree and selecting **Start PowerShell**. This starts a PowerShell session that has the SQL Server PowerShell support enabled, and the path set to the object you

right-clicked in Object Explorer. You can then enter PowerShell commands in an interactive PowerShell environment. For more information, see SQL Server PowerShell.

Related content

- Object Explorer
- Open and Configure Object Explorer
- Connect to a SQL Server or Azure SQL Database
- Object Explorer Details Pane
- Custom Reports in Management Studio

Feedback

Was this page helpful? Yes No

Use Custom Reports with Object Explorer Node Properties

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Custom reports can execute in the context of a selected Object Explorer node if the custom reports reference the report parameters of that node. This enables a custom report to use the current context, for example the current database, or a database or server object.

Object Explorer Node Report Parameters

Expand table

Parameter name	Data type
ObjectName	String
ObjectTypeName	String
Filtered	Boolean
ServerName	String
FontName	String
DatabaseName	String

Object Explorer Node Report Parameters Example

To run the example, use the following procedure.

To view the report parameter values for a node in Object Explorer

1. Copy the following sample code into a new text file and name the file by using an .rdl extension.

2. Copy the report file to a folder that you have created on the database server for custom reports.

3. In Management Studio, right-click a node in Object Explorer, point to **Reports**, click Custom Reports. In the **Open File** dialog box, locate the custom reports folder and select the report file, and then click **Open**.

   When a new custom report is first opened from an Object Explorer node, the custom report is added to the most recently used list under **Custom Reports** on the shortcut menu of that node. When a standard report is opened for the first time, it will also appear on the most recently used list under **Custom Reports**. If a custom report file is deleted, the next time that the item is selected, a prompt will appear to delete the item from the most recently used list.

   a. To change the number of files displayed on the recently used list, on the **Tools** menu, click **Options**, expand the **Environment** folder, and then click **General**.

   b. Adjust the number for **Display files in recently used list**.

Custom Report Code Sample

The report that is created by using the following code will use the parameters that are associated with an Object Explorer node.

```
<?xml version="1.0" encoding="utf-8"?>
<Report
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2005/01/reportdefini
tion"
xmlns:rd="http://schemas.microsoft.com/SQLServer/reporting/reportdesigner">
<ReportParameters>
<ReportParameter Name="ObjectName">
<DataType>String</DataType>
<Nullable>true</Nullable>
<AllowBlank>true</AllowBlank>
<Prompt>ObjectName</Prompt>
</ReportParameter>
<ReportParameter Name="ObjectTypeName">
<DataType>String</DataType>
<Nullable>true</Nullable>
<AllowBlank>true</AllowBlank>
<Prompt>ObjectTypeName</Prompt>
</ReportParameter>
<ReportParameter Name="Filtered">
<DataType>Boolean</DataType>
<Nullable>true</Nullable>
<AllowBlank>true</AllowBlank>
<Prompt>Filtered</Prompt>
</ReportParameter>
<ReportParameter Name="ServerName">
```

```
<DataType>String</DataType>
<Nullable>true</Nullable>
<AllowBlank>true</AllowBlank>
<Prompt>ServerName</Prompt>
</ReportParameter>
<ReportParameter Name="FontName">
<DataType>String</DataType>
<DefaultValue>
<Values>
<Value>Tahoma</Value>
</Values>
</DefaultValue>
<AllowBlank>true</AllowBlank>
<Prompt>FontName</Prompt>
</ReportParameter>
<ReportParameter Name="DatabaseName">
<DataType>String</DataType>
<Nullable>true</Nullable>
<DefaultValue>
<Values>
<Value>master</Value>
</Values>
</DefaultValue>
<AllowBlank>true</AllowBlank>
<Prompt>DatabaseName</Prompt>
</ReportParameter>
</ReportParameters>
<DataSources>
<DataSource Name="AllReportParameters">
<ConnectionProperties>
<IntegratedSecurity>true</IntegratedSecurity>
<ConnectString>Data Source=.</ConnectString>
<DataProvider>SQL</DataProvider>
</ConnectionProperties>
<rd:DataSourceID>f1feee4c-0fdc-4301-9efa-3cd89eed2d9f</rd:DataSourceID>
</DataSource>
</DataSources>
<BottomMargin>1in</BottomMargin>
<RightMargin>1in</RightMargin>
<rd:DrawGrid>true</rd:DrawGrid>
<InteractiveWidth>8.5in</InteractiveWidth>
<rd:SnapToGrid>true</rd:SnapToGrid>
<Body>
<ReportItems>
<Textbox Name="textbox1">
<rd:DefaultName>textbox1</rd:DefaultName>
<ZIndex>1</ZIndex>
<Width>6in</Width>
<Style>
<PaddingLeft>2pt</PaddingLeft>
<PaddingBottom>2pt</PaddingBottom>
<FontFamily>Tahoma</FontFamily>
<FontWeight>700</FontWeight>
<FontSize>20pt</FontSize>
<Color>SteelBlue</Color>
```

```
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>
</Style>
<CanGrow>true</CanGrow>
<Height>0.36in</Height>
<Value>AllReportParameters</Value>
</Textbox>
<Table Name="table1">
<DataSetName>AllReportParameters</DataSetName>
<Top>0.36in</Top>
<Details>
<TableRows>
<TableRow>
<TableCells>
<TableCell>
<ReportItems>
<Textbox Name="ObjectName">
<rd:DefaultName>ObjectName</rd:DefaultName>
<ZIndex>5</ZIndex>
<Style>
<BorderStyle>
<Default>Solid</Default>
</BorderStyle>
<PaddingLeft>2pt</PaddingLeft>
<PaddingBottom>2pt</PaddingBottom>
<FontFamily>Tahoma</FontFamily>
<BorderColor>
<Default>LightGrey</Default>
</BorderColor>
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>
</Style>
<CanGrow>true</CanGrow>
<Value>=Parameters!ObjectName.Value</Value>
</Textbox>
</ReportItems>
</TableCell>
<TableCell>
<ReportItems>
<Textbox Name="ObjectTypeName">
<rd:DefaultName>ObjectTypeName</rd:DefaultName>
<ZIndex>4</ZIndex>
<Style>
<BorderStyle>
<Default>Solid</Default>
</BorderStyle>
<PaddingLeft>2pt</PaddingLeft>
<PaddingBottom>2pt</PaddingBottom>
<FontFamily>Tahoma</FontFamily>
<BorderColor>
<Default>LightGrey</Default>
</BorderColor>
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>
</Style>
```

```
<CanGrow>true</CanGrow>
<Value>=Parameters!ObjectTypeName.Value</Value>
</Textbox>
</ReportItems>
</TableCell>
<TableCell>
<ReportItems>
<Textbox Name="Filtered">
<rd:DefaultName>Filtered</rd:DefaultName>
<ZIndex>3</ZIndex>
<Style>
<BorderStyle>
<Default>Solid</Default>
</BorderStyle>
<PaddingLeft>2pt</PaddingLeft>
<PaddingBottom>2pt</PaddingBottom>
<FontFamily>Tahoma</FontFamily>
<BorderColor>
<Default>LightGrey</Default>
</BorderColor>
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>
</Style>
<CanGrow>true</CanGrow>
<Value>=Parameters!Filtered.Value</Value>
</Textbox>
</ReportItems>
</TableCell>
<TableCell>
<ReportItems>
<Textbox Name="ServerName">
<rd:DefaultName>ServerName</rd:DefaultName>
<ZIndex>2</ZIndex>
<Style>
<BorderStyle>
<Default>Solid</Default>
</BorderStyle>
<PaddingLeft>2pt</PaddingLeft>
<PaddingBottom>2pt</PaddingBottom>
<FontFamily>Tahoma</FontFamily>
<BorderColor>
<Default>LightGrey</Default>
</BorderColor>
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>
</Style>
<CanGrow>true</CanGrow>
<Value>=Parameters!ServerName.Value</Value>
</Textbox>
</ReportItems>
</TableCell>
<TableCell>
<ReportItems>
<Textbox Name="FontName">
<rd:DefaultName>FontName</rd:DefaultName>
```

```
<ZIndex>1</ZIndex>
<Style>
<BorderStyle>
<Default>Solid</Default>
</BorderStyle>
<PaddingLeft>2pt</PaddingLeft>
<PaddingBottom>2pt</PaddingBottom>
<FontFamily>Tahoma</FontFamily>
<BorderColor>
<Default>LightGrey</Default>
</BorderColor>
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>
</Style>
<CanGrow>true</CanGrow>
<Value>=Parameters!FontName.Value</Value>
</Textbox>
</ReportItems>
</TableCell>
<TableCell>
<ReportItems>
<Textbox Name="DatabaseName">
<rd:DefaultName>DatabaseName</rd:DefaultName>
<Style>
<BorderStyle>
<Default>Solid</Default>
</BorderStyle>
<PaddingLeft>2pt</PaddingLeft>
<PaddingBottom>2pt</PaddingBottom>
<FontFamily>Tahoma</FontFamily>
<BorderColor>
<Default>LightGrey</Default>
</BorderColor>
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>
</Style>
<CanGrow>true</CanGrow>
<Value>=Parameters!DatabaseName.Value</Value>
</Textbox>
</ReportItems>
</TableCell>
</TableCells>
<Height>0.21in</Height>
</TableRow>
</TableRows>
</Details>
<Header>
<TableRows>
<TableRow>
<TableCells>
<TableCell>
<ReportItems>
<Textbox Name="textbox2">
<rd:DefaultName>textbox2</rd:DefaultName>
<ZIndex>11</ZIndex>
```

```
<Style>
<BorderStyle>
<Default>Solid</Default>
</BorderStyle>
<PaddingLeft>2pt</PaddingLeft>
<PaddingBottom>2pt</PaddingBottom>
<FontFamily>Tahoma</FontFamily>
<FontWeight>700</FontWeight>
<FontSize>11pt</FontSize>
<BorderColor>
<Default>LightGrey</Default>
</BorderColor>
<BackgroundColor>SteelBlue</BackgroundColor>
<Color>White</Color>
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>
</Style>
<CanGrow>true</CanGrow>
<Value>ObjectName</Value>
</Textbox>
</ReportItems>
</TableCell>
<TableCell>
<ReportItems>
<Textbox Name="textbox3">
<rd:DefaultName>textbox3</rd:DefaultName>
<ZIndex>10</ZIndex>
<Style>
<BorderStyle>
<Default>Solid</Default>
</BorderStyle>
<PaddingLeft>2pt</PaddingLeft>
<PaddingBottom>2pt</PaddingBottom>
<FontFamily>Tahoma</FontFamily>
<FontWeight>700</FontWeight>
<FontSize>11pt</FontSize>
<BorderColor>
<Default>LightGrey</Default>
</BorderColor>
<BackgroundColor>SteelBlue</BackgroundColor>
<Color>White</Color>
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>
</Style>
<CanGrow>true</CanGrow>
<Value>ObjectTypeName</Value>
</Textbox>
</ReportItems>
</TableCell>
<TableCell>
<ReportItems>
<Textbox Name="textbox4">
<rd:DefaultName>textbox4</rd:DefaultName>
<ZIndex>9</ZIndex>
<Style>
```

```
<BorderStyle>
<Default>Solid</Default>
</BorderStyle>
<PaddingLeft>2pt</PaddingLeft>
<PaddingBottom>2pt</PaddingBottom>
<FontFamily>Tahoma</FontFamily>
<FontWeight>700</FontWeight>
<FontSize>11pt</FontSize>
<BorderColor>
<Default>LightGrey</Default>
</BorderColor>
<BackgroundColor>SteelBlue</BackgroundColor>
<Color>White</Color>
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>
</Style>
<CanGrow>true</CanGrow>
<Value>Filtered</Value>
</Textbox>
</ReportItems>
</TableCell>
<TableCell>
<ReportItems>
<Textbox Name="textbox5">
<rd:DefaultName>textbox5</rd:DefaultName>
<ZIndex>8</ZIndex>
<Style>
<BorderStyle>
<Default>Solid</Default>
</BorderStyle>
<PaddingLeft>2pt</PaddingLeft>
<PaddingBottom>2pt</PaddingBottom>
<FontFamily>Tahoma</FontFamily>
<FontWeight>700</FontWeight>
<FontSize>11pt</FontSize>
<BorderColor>
<Default>LightGrey</Default>
</BorderColor>
<BackgroundColor>SteelBlue</BackgroundColor>
<Color>White</Color>
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>
</Style>
<CanGrow>true</CanGrow>
<Value>ServerName</Value>
</Textbox>
</ReportItems>
</TableCell>
<TableCell>
<ReportItems>
<Textbox Name="textbox6">
<rd:DefaultName>textbox6</rd:DefaultName>
<ZIndex>7</ZIndex>
<Style>
<BorderStyle>
```

```
<Default>Solid</Default>
</BorderStyle>
<PaddingLeft>2pt</PaddingLeft>
<PaddingBottom>2pt</PaddingBottom>
<FontFamily>Tahoma</FontFamily>
<FontWeight>700</FontWeight>
<FontSize>11pt</FontSize>
<BorderColor>
<Default>LightGrey</Default>
</BorderColor>
<BackgroundColor>SteelBlue</BackgroundColor>
<Color>White</Color>
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>
</Style>
<CanGrow>true</CanGrow>
<Value>FontName</Value>
</Textbox>
</ReportItems>
</TableCell>
<TableCell>
<ReportItems>
<Textbox Name="textbox7">
<rd:DefaultName>textbox7</rd:DefaultName>
<ZIndex>6</ZIndex>
<Style>
<BorderStyle>
<Default>Solid</Default>
</BorderStyle>
<PaddingLeft>2pt</PaddingLeft>
<PaddingBottom>2pt</PaddingBottom>
<FontFamily>Tahoma</FontFamily>
<FontWeight>700</FontWeight>
<FontSize>11pt</FontSize>
<BorderColor>
<Default>LightGrey</Default>
</BorderColor>
<BackgroundColor>SteelBlue</BackgroundColor>
<Color>White</Color>
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>
</Style>
<CanGrow>true</CanGrow>
<Value>DatabaseName</Value>
</Textbox>
</ReportItems>
</TableCell>
</TableCells>
<Height>0.22in</Height>
</TableRow>
</TableRows>
<RepeatOnNewPage>true</RepeatOnNewPage>
</Header>
<TableColumns>
<TableColumn>
```

```
<Width>3in</Width>
</TableColumn>
<TableColumn>
<Width>1.5in</Width>
</TableColumn>
<TableColumn>
<Width>0.75in</Width>
</TableColumn>
<TableColumn>
<Width>1.125in</Width>
</TableColumn>
<TableColumn>
<Width>2in</Width>
</TableColumn>
<TableColumn>
<Width>1.375in</Width>
</TableColumn>
</TableColumns>
</Table>
</ReportItems>
<Height>0.79in</Height>
</Body>
<rd:ReportID>abb14e58-fd36-495a-89ff-b66f29ef287b</rd:ReportID>
<LeftMargin>1in</LeftMargin>
<DataSets>
<DataSet Name="AllReportParameters">
<Query>
<rd:UseGenericDesigner>true</rd:UseGenericDesigner>
<CommandText>SELECT 1
</CommandText>
<DataSourceName>AllReportParameters</DataSourceName>
</Query>
<Fields>
<Field Name="ObjectName">
<rd:TypeName>System.String</rd:TypeName>
<DataField>ObjectName</DataField>
</Field>
<Field Name="ObjectTypeName">
<rd:TypeName>System.String</rd:TypeName>
<DataField>ObjectTypeName</DataField>
</Field>
<Field Name="Filtered">
<rd:TypeName>System.Boolean</rd:TypeName>
<DataField>Filtered</DataField>
</Field>
<Field Name="ServerName">
<rd:TypeName>System.String</rd:TypeName>
<DataField>ServerName</DataField>
</Field>
<Field Name="FontName">
<rd:TypeName>System.String</rd:TypeName>
<DataField>FontName</DataField>
</Field>
<Field Name="DatabaseName">
<rd:TypeName>System.String</rd:TypeName>
```

```
<DataField>DatabaseName</DataField>
</Field>
</Fields>
</DataSet>
</DataSets>
<Width>6.875in</Width>
<InteractiveHeight>11in</InteractiveHeight>
<Language>en-US</Language>
<TopMargin>1in</TopMargin>
</Report>
```

See Also

Custom Reports in Management Studio
Add a Custom Report to Management Studio
Unsuppress Run Custom Report Warnings

Feedback

Was this page helpful? Yes No

Object Dependencies

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Some database objects have dependencies upon other database objects. For example, views and stored procedures depend upon the existence of tables that contain the data returned by the view or procedure. The **Object Dependencies (General Page)** for the current object lists both the database objects that must be present for the object to function properly and the objects that depend upon the selected object. An object that references another object in its definition and that definition is stored in the system catalog is called a *referencing entity*. An object that is referred to by another object is called a *referenced entity*.

The **Object Dependencies (Advanced Page)** for the current object lists the SQL Server database objects and Integration Services objects that depend on the object. The objects may be stored on different servers.

Use this dialog box to understand the dependencies before changing or deleting the selected object.

UI element list

Objects that depend on <*selected object*>
Clicking this button displays a list of those objects that are dependency-tracked and that depend on the selected object.

Objects on which <*selected object*> **depends**
Clicking this button displays a list of those objects that are dependency-tracked, on which the selected object depends.

Dependencies
If **Objects that depend on** <*selected object*> is clicked, this displays an hierarchical view of objects that depend on the selected object. If **Objects on which** <*selected object*> **depends** is clicked, this displays an hierarchical view of objects on which the selected object depends.

Name
Displays the name of the object selected in the **Dependencies** tree view above.

Type
Displays the type of the object selected in the **Dependencies** tree view above.

Last Sync Time

> ⓘ **Note**
>
> This option is available only on the **Advanced** page.

Specifies the time and date when the dependency information was last updated.

Dependency type

> ⓘ **Note**
>
> This option is available only on the **General** page.

Displays the type of dependency between two objects. Can be one of the following:

- Schema-bound dependency

  Is a relationship between two objects that prevents the referenced object from being dropped or modified as long as the referencing object exists. A schema-bound dependency is created when a view or user-defined function is created by using the WITH SCHEMABINDING clause, or when a table references another object through a CHECK or DEFAULT constraint or in the definition of a computed column.

- Non-schema-bound dependency

  Is a relationship between two objects that does not prevent the referenced object from being dropped or modified.

- Not available or Unresolved Entity

  Indicates the dependency type cannot be determined. This occurs only when the selected object is on an instance of SQL Server that is earlier than SQL Server 2008 (10.0.x).

Feedback

Was this page helpful? Yes No

Hide System Objects in Object Explorer

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes how to hide system objects in Object Explorer in SQL Server by using SQL Server Management Studio. The **Databases** node of Object Explorer contains system objects such as the system databases. Use the **Tools/Options** pages to hide the system objects. Some system objects, such as system functions and system data types, are not affected by this setting.

Using SQL Server Management Studio

To hide system objects in Object Explorer

1. On the **Tools** menu, click **Options**.
2. On the **Environment/Startup** page, select **Hide system objects in Object Explorer**, and then click **OK**.
3. In the **SQL Server Management Studio** dialog box, click **OK** to acknowledge that SQL Server Management Studio must be restarted for this change to take effect.
4. Close and reopen SQL Server Management Studio.

Feedback

Was this page helpful? Yes No

Object Explorer Details Pane

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Object Explorer Details, a component of SQL Server Management Studio, provides a tabular view of all the objects in the server and presents a user interface to manage them. The capabilities of Object Explorer vary slightly depending on the type of server, but generally include the development features for databases and management features for all server types.

The Object Explorer Details pane is visible in the Management Studio by default. If you cannot see Object Explorer, on the **View** menu, click **Object Explorer Details** or press **F7**.

> **Note**
>
> Management Studio presents dates formatted with the Microsoft Windows Regional and Language Options in effect when Management Studio was started. Restart Management Studio to reflect newer settings.

Object Explorer Details

Object Explorer Details can be used to navigate through folders and objects on your SQL Server instance. On 32-bit operating systems, Object Explorer can only display 64,000 objects. An icon must be selected to access additional objects.

Object Explorer Details includes a toolbar which contains the icons described in the following table. Icons are only available when appropriate.

Expand table

Icon	Action
Back	Moves to the previous items displayed in Object Explorer Details. Re-runs a search when the previous display is the result of a search operation.
Forward	Moves to the next screen after a **Back** operation is selected.
Up	Moves to the parent object or folder.
Synchronize	Sets the focus of Object Explorer to the selected object in Object Explorer Details.

Icon	Action
Filter	When available, shows a configurable subset of objects.
Refresh	Refreshes the display in Object Explorer Details.
Search	Provides an area to enter a search term for certain database objects.

Column Header Selections

Object Explorer Details has selectable columns. You can right-click in any column header and check the items that you want to display. Your selections will be persisted across the different objects you navigate through. Selections for each user are retained when you leave and restart SQL Server Management Studio.

> ⊗ **Caution**
>
> Showing all columns for some object types (such as Databases) might slow the display rendering slightly for large sets of objects.

Sorting

Clicking one time on a column header will sort by that column. Clicking again on the same header reverse-sorts by that column. Sort selections are maintained for each user across objects and folders, and on Management Studio restart.

Filtering

Certain lists of objects displayed in Object Explorer Detail can be filtered using the **Filter** icon on the Object Explorer Details toolbar. The icon will be enabled when filtering is possible.

Details Pane

The very bottom area of Object Explorer Details contains a panel that displays certain details of the selected object. Multiple object selections show only the count of the objects. When items are selected in the panel, click the **Copy** icon to copy the displayed text to the clipboard.

The down arrow key moves the selection to the next item in the list. The up arrow key moves the selection to the previous item in the list. Holding the arrow key down will

speed through the items. The selection stops at the bottom or top of the property list.

Typing the first character of a property name moves the selection to the next property that begins with that character.

When properties are arranged in multiple columns, use the left arrow and right arrow keys to move to adjacent columns.

To copy property values to the clipboard, use the following keyboard key combinations:

- Ctrl+C copies the property value.
- Ctrl+Shift+C copies the property name and the property value with a tab delimiter.

Use the right-click menu in the Object Explorer Details property pane to copy all properties or all properties and names to the clipboard.

To display a property value when the field width does not completely display a property value, hover the mouse cursor over the value or set UI focus on the property value to activate a tool tip with the entire property value. Accessibility screen readers will report the full property value when the user focuses on the long property value.

If the number of properties in the property pane exceeds that which will fit in the content area, a scroll bar will appear on the right side of the property pane. Use the scroll bar to adjust the view of property values in the content area.

Multiple Object Selection

Object Explorer Details supports multiple object selection. For example, if you select Tables in Object Explorer, then in the Object Explorer Details window if you hold down the **Ctrl** key, you can select several tables, right-click them, and then select **Delete** or **Script Table AS** to act on all the selected tables immediately. The standard copy commands will copy the displayed text to the clipboard.

SQL Server Object Search

Wildcards

- Standard wildcard characters are supported. For example, searching for **dm_os%counters** returns both dm_os_memory_cache_counters and dm_os_performance_counters. For more information, see How to: Search with regular expressions.

Search Scope

- Each search is scoped to the current focus in the Object Explorer tree. For example, if the Object Explorer focus is on a database, search for **dm_os%counters** returns dm_os_memory_cache_counters and dm_os_performance_counters in that database. If the Object Explorer focus is on the Databases node, all databases are searched and multiple instances of the dynamic views are returned.

Large Sets

- Searching large object sets can take a long time and slow server performance.

See Also

Object Explorer

Feedback

Was this page helpful? Yes No

Open and Configure Object Explorer

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Open Object Explorer by using the **View** menu in SQL Server Management Studio. Configure Object Explorer options by using **Options** on the **Tools** menu.

Viewing Object Explorer

Object Explorer is visible in Management Studio by default. If you cannot see Object Explorer, on the **View** menu, click **Object Explorer**.

Configuring Object Explorer Options

To configure the Object Explorer settings, on the **Tools** menu, click **Options**, and then click **SQL Server Object Explorer**.

See Also

Object Explorer
Connect to an Instance From Object Explorer
Manage Objects by Using Object Explorer
Object Explorer Details Pane
Custom Reports in Management Studio

Feedback

Was this page helpful? Yes No

Connect to a SQL Server or Azure SQL Database

Article • 02/13/2025

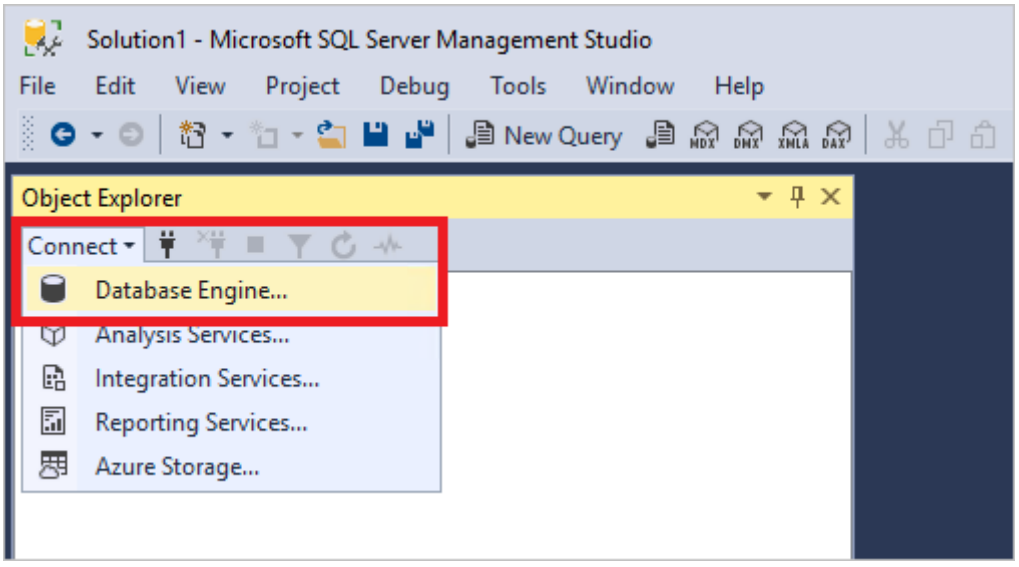
Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

To work with servers and databases, you must first connect to the server. You can connect to multiple servers at the same time.

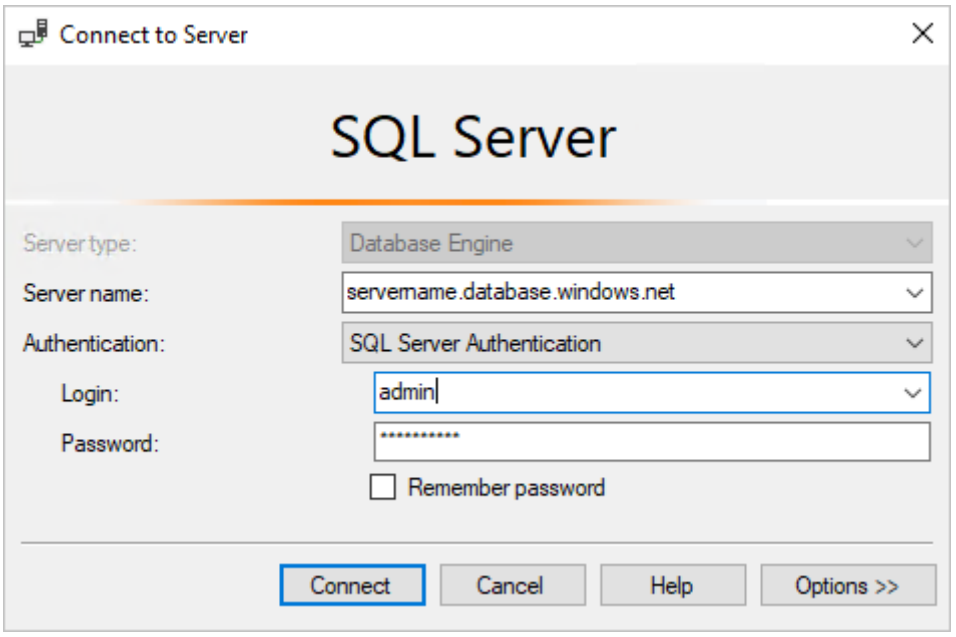
SQL Server Management Studio (SSMS) supports several types of connections. This article provides details for connecting to SQL Server and Azure SQL Database (connecting to an Azure SQL single database or elastic pool). For information on the other connection options, see the links at the bottom of this page.

Connecting to a Server

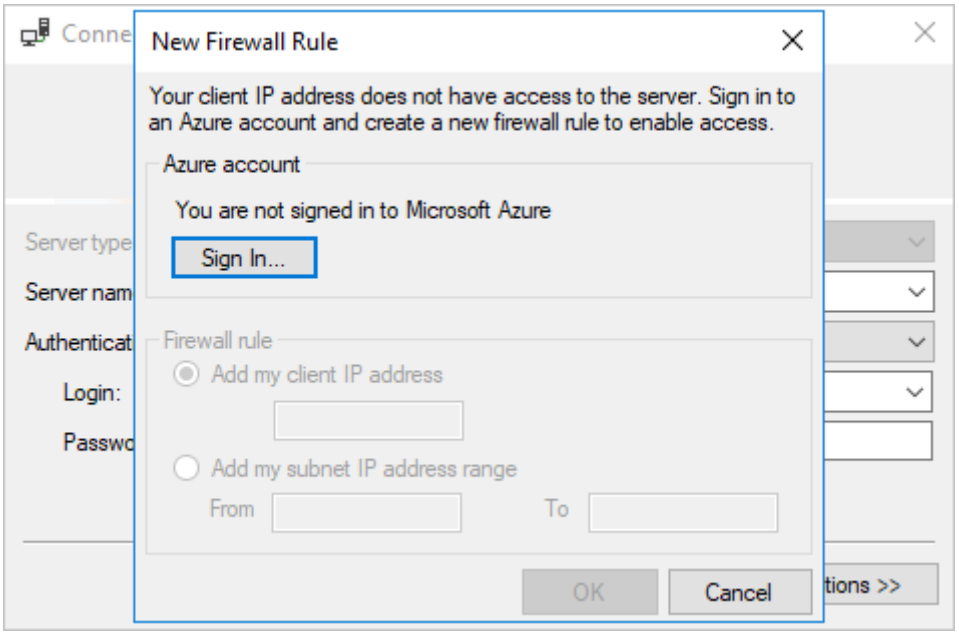
1. On **Object Explorer**, click **Connect** > **Database Engine...**.



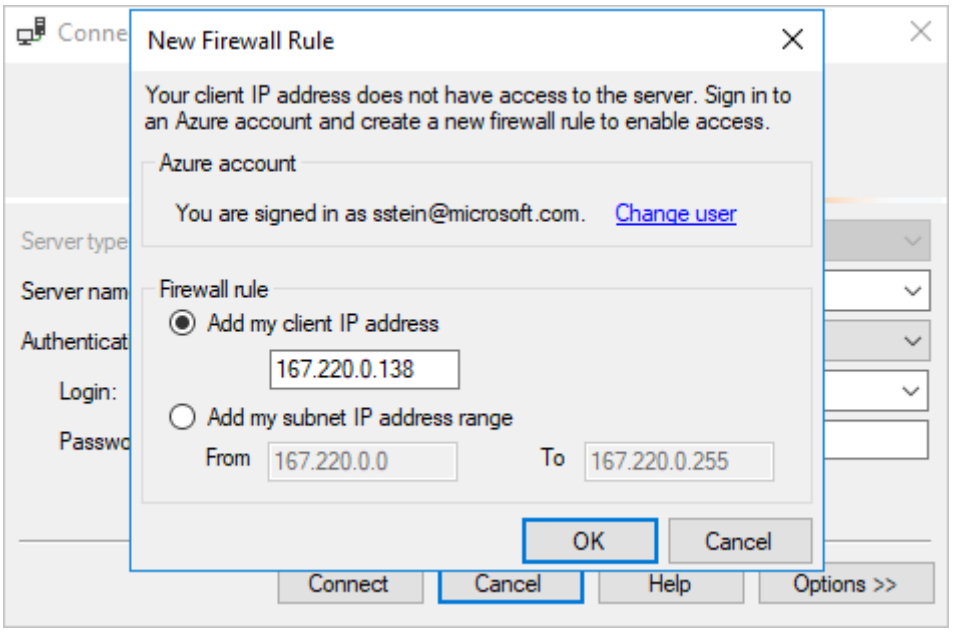
2. Fill out the **Connect to Server** form and click **Connect**:



3. If you're connecting to an Azure SQL Server, you might get prompted to sign in to create a firewall rule. Click **Sign In...** (if not, skip to step 6 below)

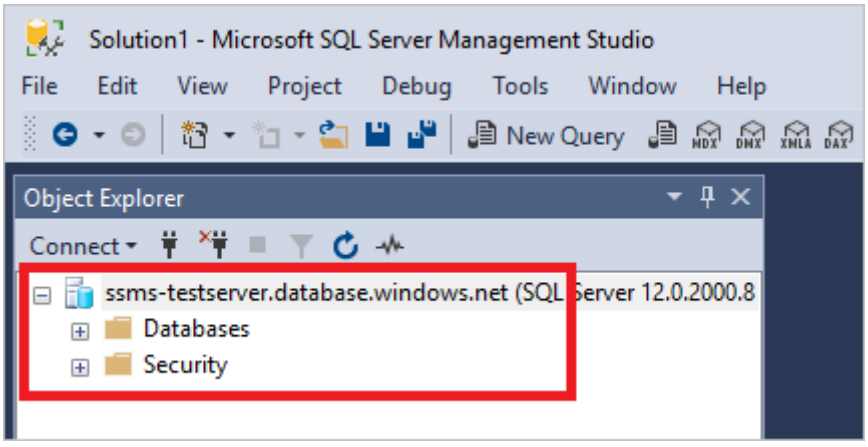


4. After successfully signing in, the form is pre-populated with your specific IP address. If your IP address changes often, it might be easier to grant access to a range, so select the option that's best for your environment.



5. To create the firewall rule and connect to the server, click **OK**.

6. The server appears in **Object Explorer** after successfully connecting:



Next Steps

Design, Create, and Update Tables

See Also

SQL Server Management Studio (SSMS)
Download SQL Server Management Studio (SSMS)

Analysis Services
Integration Services
Reporting Services
Azure Storage

Feedback

Was this page helpful? Yes No

Browse for Objects (Object Explorer)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Displays a list of objects of the selected type. Select one of more of the objects from this list by selecting the corresponding check box. The **Browse for Objects** box only displays items appropriate for the action, and only displays items that you have permission to view.

Options

Check box
Select to include the object.

Icon
An icon representing the object type.

Name
The name of the object.

Type
The type of database object.

Feedback

Was this page helpful? Yes No

Select Object Types (Object Explorer)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use this dialog box to specify the types of objects that appear in the **Select Objects** dialog box.

Select the Types of Objects to Find

Select items in the **Object Type** list. The list contains only the items available for the type of action; for instance, only **Logins** appear if you are selecting the owner for a new object.

Feedback

Was this page helpful? Yes No

Select Objects (Object Explorer)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use the **Select Objects** dialog box to add an object to a list in another dialog box. The dialog box title and the options available in the dialog box depend upon how it was opened. Only available options appear; for instance, only logins are available when you are selecting an owner for a new object.

Options

Select these object types
Displays a list of the types of which the objects to be selected belong. The types include Microsoft SQL Server level and database level principals and securables. This box is populated from the selections made from the **Select Object Types** dialog box, accessed from the **Objects Type** button.

Enter the object names to select
Enter a semicolon-separated list of names of the objects to be selected. Objects to be selected must be of a type listed in the **Select these object types** box. Objects can be selected from a list accessed by clicking the **Browse** button.

Object Types
Displays a list of object types. Select one or more by selecting the check box corresponding to the type.

Check Names
Validates the object names in the **Enter the object names to select** box. If an object name that is not valid is listed, the **Name not Found** dialog box is presented. With this dialog box, the name can be corrected or removed from the list of objects to select.

Browse
Presents the **Browse for Objects** dialog box. This contains a list of objects of the types listed in the **Select These Object Types** box. Select objects from this list by selecting the corresponding check boxes.

Feedback

Was this page helpful? Yes No

Delete Objects

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Use this dialog box to delete a database or database object.

UI element list

Object to be deleted
Displays the names, types, owners, and status of the objects that are about to be deleted, as well as any messages about errors during execution.

> **Note**
>
> Running **Delete** on a database is equivalent to issuing DROP DATABASE in Transact-SQL.

Show Dependencies
Click to display both the objects that are dependent on the currently selected object and objects that the current object is dependent on (upward and downward dependency). The information displayed in the **Show Dependencies** dialog box is read-only.

> **Note**
>
> The **Show Dependencies** button does not appear for all types of database objects. To view dependencies when the **Show Dependencies** button is not available, right-click the object in Object Explorer, and then click **View Dependencies**.

Delete backup and restore history information for databases
Only appears when a database is deleted, this check box causes the backup and restore history for the subject database to be deleted from the **msdb** database.

Close existing connections
Only appears when a database is deleted, this check box terminates connections to the subject database.

Feedback

Was this page helpful? Yes No

Add a Custom Report to Management Studio

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes how to create a simple Reporting Services report that is saved as an .rdl file, and then add that rdl file to SQL Server Management Studio as a custom report. SSRS can create a wide variety of sophisticated reports. To create a report by using this topic, you must have SQL Server Data Tools (SSDT) installed on the computer. You do not have to install SSRS on SQL Server to run a custom report using Management Studio.

To create a simple report saved as an .rdl file

1. Click **Start**, point to **Programs**, point to **Microsoft SQL Server**, and then click **SQL Server Data Tools**.
2. On the **File** menu, point to **New**, and then click **Project**.
3. In the **Project Types** list, click **Business Intelligence Projects**.
4. In the **Templates** list, click **Report Server Project Wizard**.
5. In **Name**, type **ConnectionsReport**, and then click **OK**.
6. On the **Report Wizard** introduction page, click **Next**.
7. On the **Select the Data Source** page, in the Name box type a name for this connection to your Database Engine, and then click **Edit**.
8. In the **Connection Properties** dialog box, in the **Server name** box, type the name of your instance of the Database Engine.
9. In the **Select or enter a database name** box, type the name of any database on your SQL Server, such as `AdventureWorks2022`, and then click **OK**.
10. On the **Select the Data Source** page, click **Next**.
11. On the **Design the Query** page, in the **Query string** box, type the following Transact-SQL statement that lists the current connections to your Database Engine,

and then click **Next**. The Report Wizard Query string box will not accept report parameters. More complex custom reports must be created manually.

SELECT session_id, net_transport FROM sys.dm_exec_connections;

12. On the **Select the Report Type** page, select **Tabular**, and then click **Finish**.
13. On the **Completing the Wizard** page, in the **Report name** box, type **ConnectionsReport**, and then click **Finish** to create and save the report.
14. Close SQL Server Data Tools.
15. Copy **ConnectionsReport.rdl** to a folder that you created on the database server for custom reports.

To add a report to Management Studio

- In Management Studio, right-click a node in Object Explorer, point to **Reports**, click **Custom Reports**. In the **Open File** dialog box, locate the custom reports folder and select the **ConnectionsReport.rdl** file, and then click **Open**.

  When a new custom report is first opened from an Object Explorer node, the custom report is added to the most recently used list under **Custom Reports** on the shortcut menu of that node. When a standard report is opened for the first time, it will also appear on the most recently used list under **Custom Reports**. If a custom report file is deleted, the next time that the item is selected, a prompt will appear to delete the item from the most recently used list.

  1. To change the number of files that are displayed on the recently used list, on the **Tools** menu, click **Options,** expand the **Environment** folder, and then click **General**.
  2. Adjust the number for **Display files in recently used list**.

See Also

Custom Reports in Management Studio
Use Custom Reports with Object Explorer Node Properties
Unsuppress Run Custom Report Warnings
SQL Server Reporting Services

Feedback

Was this page helpful? Yes No

Custom Reports in Management Studio

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

In SQL Server Management Studio, many Object Explorer nodes display a set of standard reports that are created by Microsoft. These reports summarize typically requested server information. Starting with SQL Server 2005 (9.x) Service Pack 2, administrators can run custom reports that were created in SQL Server Data Tools (SSDT) from Management Studio.

Implementation

Custom reports are stored as report definition (.rdl) files and are created by using Report Definition Language (RDL). RDL contains data retrieval and layout information for a report in an XML format. RDL is an open schema. Developers can extend RDL with additional attributes and elements. Reports can execute any valid Transact-SQL statement within the report.

If Object Explorer is connected to a server, custom reports can execute in the context of the current Object Explorer selection if the reports reference report parameters of that node. This enables a report to use the current context, such as the current database; or a consistent context, such as specifying a designated database as part of the Transact-SQL statement that is contained in the custom report.

Running a Custom Report

You can run a custom report in Management Studio in the following ways:

- Right-click a node in Object Explorer, point to **Reports** and left-click **Custom Reports**. In the **Open File** dialog box, locate a folder that contains .rdl files, and then open the appropriate report file.
- Right-click a node in Object Explorer, point to **Reports**, point to **Custom Reports**, and then select a custom report from the most recently used file list.

Limitations

When you work with custom reports, consider the following limitations:

- To prevent the unintended execution of malicious code, Management Studio cannot be configured to automatically run a report, even if the file system is configured to associate .rdl files with Management Studio. Reports cannot be programmatically executed in Management Studio and cannot run from the command line through Management Studio.
- You can run custom reports in a context that does not produce the expected values. For example, you can run a report about replication in the context of a database that is not involved in replication, or run a report as a user who does not have permission to access information that is required to generate an accurate report. The creator of the custom report is responsible for the validity of the report structure and its context.
- You cannot add a custom report to the list of standard reports.
- The code processed by the report might affect server performance.
- Custom reports will not support subreports.
- The command text for each query within the report must not be defined through an expression.
- Any query parameter that is used in a command (query) can only reference a single report parameter and cannot use any expression operators.
- Only Text and Stored Procedure command types are supported for report commands (queries).
- The report framework does not provide any parameter escaping for the queries. Query authors must make sure that their queries are free from SQL injection attacks.

Managing Custom Reports

We recommend that users who have many custom reports organize them by using file system folders that have appropriate NTFS file system permissions.

Permissions

Custom reports run by using the permissions of the current user. To prevent a malicious user from changing the queries run by the report, permissions on the file system folder that contains the report files should be set to restrict access.

Both the user and the account that is used by the SQL Server service require read access to the file system folder that contains the report files.

Any valid .NET Framework command can be embedded in a report, but the command will not be executed.

> **⊗ Caution**
>
> Any valid Transact-SQL statement can be embedded in and executed from a report. Running a report under a high-privileged user account makes it possible for any of these embedded instructions to execute without challenge.

See Also

Add a Custom Report to Management Studio
Unsuppress Run Custom Report Warnings
Use Custom Reports with Object Explorer Node Properties

Feedback

Was this page helpful? Yes No

Add Objects

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Limits the object selection dialog boxes to the types of objects you want to see.

Options

Specific objects
Open the **Select Objects** dialog box for you to further define your search for objects.

All objects of the types
Open the **Select Object Types** dialog box to specify the type of objects that should be included in the underlying list.

The server *<servername>*
Add all appropriate objects on the specified server. Only appears when opened from server scoped dialog boxes such as the **Login - New** dialog box.

All objects belonging to the schema
Add all objects owned by the schema specified in the **Schema name** box. Only appears for database scoped dialog boxes such as the **Database User - New** dialog box.

Feedback

Was this page helpful? Yes No

Unsuppress Run Custom Report Warnings

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

There are two warning dialog boxes for custom reports. This topic describes how to unsuppress the display of these boxes in SQL Server by using SQL Server Management Studio.

By default, the **Run Custom Reports** dialog box appears before a custom report runs. If you select the **Please don't show this warning again** check box, the dialog box will no longer appear. Also by default, the **Run Custom Reports** dialog box appears when you open a custom report and then click a link to open another custom report. This dialog box displays the fill path to the drill-through custom report file. If you select the **Please don't show this warning again** check box, the dialog box will no longer appear.

Using SQL Server Management Studio

To unsuppress the main custom report warning dialog box

1. Connect to `<server>\<share or drive>\Documents and Settings\<UserProfile>\Application Data\Microsoft\Microsoft SQL Server\130\Tools\Shell\reports.xml`.
2. Right-click **reports.xml**, and then click **Edit**.
3. Change **<SuppressWarning>true</SuppressWarning>** to **<SuppressWarning>false</SuppressWarning>**.
4. Restart SQL Server Management Studio.

To unsuppress the drill-through custom report warning dialog box

1. Connect to `<server>\<share or drive>\Documents and Settings\<UserProfile>\Application Data\Microsoft\Microsoft SQL Server\130\Tools\Shell\reports.xml`.
2. Right-click **reports.xml**, and click **Edit**.

3. Change **<SuppressDrillthroughWarning>true</SuppressDrillthroughWarning>** to **<\SuppressDrillthroughWarning>false</SuppressDrillthroughWarning>**.

4. Restart SQL Server Management Studio.

See Also

Custom Reports in Management Studio
Add a Custom Report to Management Studio
Use Custom Reports with Object Explorer Node Properties

Feedback

Was this page helpful? Yes No

Solution Explorer

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The Solution Explorer pane in Microsoft SQL Server Management Studio provides containers called projects for managing items such as database scripts, queries, data connections, and files. One or more projects that are related to each other can be combined in a container called a solution.

A solution includes one or more projects, plus files and metadata that help define the solution as a whole. A project is a set of files, plus related metadata such as connection information. Solutions and projects contain items that represent the scripts, queries, connection information and files that you need to create your database solution.

Benefits

Use these containers to:

- Implement source control on queries and scripts.
- Manage settings for your solution as a whole or for individual projects.
- Handle the details of file management while you focus on items that make up your database solution.
- Add items that are useful to multiple projects in the solution or to the solution without referencing the item in each project.
- Work on miscellaneous files that are independent from solutions or projects.

The items contained in projects depend on the project type and whether you are using SQL Server Management Studio.

Related tasks

Use the following topics to get started with SQL Server Solutions:

Expand table

Description	Topic
Describes how to collect one or more projects in a solution.	Solutions (SQL Server Management Studio)
Describes how to create a project and add items like scripts and connections.	Projects (SQL Server Management Studio)
Provides information about the files used by SQL Server Management Studio to manage solutions and files.	Files That Manage Solutions and Projects

Feedback

Was this page helpful? Yes No

Copy Items in a Solution

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can copy items using Solution Explorer or Windows Explorer.

To copy items within Solution Explorer

1. In Solution Explorer, select the item you want to copy.
2. On the **Edit** menu, click **Copy**.
3. In Solution Explorer, select the destination project.
4. On the **Edit** menu, click **Paste**.

> **Note**
>
> Connections cannot be copied between projects. When copying a query with an associated connection to another project, its associated connection is not copied to the target project. You must manually create the connection in the target project.

See Also

Solution Explorer
Move Items in a Solution

Feedback

Was this page helpful? Yes No

Files That Manage Solutions and Projects

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

This topic describes the file types that are specific to Microsoft SQL Server Management Studio. By default, all solutions and their projects are created in \My Documents\SQL Server Management Studio Projects.

Management Studio Solution Files

SQL Server Management Studio uses different file types than Microsoft SQL Server Data Tools (SSDT) or Microsoft Visual Studio. This means you cannot open a SQL Server Management Studio solution in SQL Server Data Tools (SSDT) or in Visual Studio. SQL Server Management Studio solution files allow Solution Explorer to display a graphical interface for managing your files.

Expand table

Extension	File type	Description	Created by
.ssmssln	SQL Server Management Studio Solution Object	Provides the environment with references to the location on disk of Microsoft SQL Server projects, project items, and solution	SQL Server Management Studio

Management Studio Project Files

In the same way that solutions contain solution files that manage objects in a solution, projects contain project files. The type of project file that SQL Server Management Studio creates for a project depends on the template used to create the project. The following table describes the type of file created for each project.

Expand table

Extension	Project template
.ssmssqlproj	SQL Server Scripts Project

Extension	Project template
.ssmsasproj	Analysis Services Scripts Project

Location of Solution-Level Files

By default, solution-level files are created in the physical directory of the first project that is created with the solution. You can specify a directory for the solution by creating a solution, or you can specify the directory when you create a new project.

If you have a directory structure similar to the logical structure shown in Solution Explorer, project and solution files are easier to locate and share with other developers on a team.

See Also

Manage Files with Encoding
Miscellaneous Files
Solution Explorer
Solutions (SQL Server Management Studio)
Projects (SQL Server Management Studio)
Solution Explorer Source Control

Feedback

Was this page helpful? Yes No

Solutions (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

A SQL Server Management Studio solution is a collection of one or more related projects. Projects are containers that developers use to organize related files, such as sets of commonly used administration scripts.

Solution Overview

You can use Management Studio as a script development platform for Database Engine and Analysis Services. Use the Management Studio code editors to develop scripts and queries for relational and multidimensional databases, and collect related scripts and queries together in projects.

Projects can contain:

- Queries and scripts that implement production processes.
- Connection information and files used by the queries and scripts.

One or more related projects can be combined in a solution. Solutions and projects can be managed by using the Solution Explorer pane in Management Studio.

Solutions and projects can be integrated into a Microsoft Visual SourceSafe (VSS) database or other third-party source control providers, for development change tracking and life-cycle management.

Solution Tasks

Expand table

Task Description	Topic
Describes how to create a new solution that can then be used to contain one or more projects.	Create a New Solution
Describes how to open an existing solution in Solution Explorer.	Open an Existing Solution

Task Description	Topic
Describes how to close a solution.	Close a Solution
Describes how to delete a solution.	Delete a Solution
Describes how to copy items between projects.	Copy Items in a Solution
Describes how to delete items in a project, or an entire project.	Remove or Delete an Item or Project
Describes how to move items between projects in a solution.	Move Items in a Solution
Describes how to rename a solution or the items in the solution.	Rename Solutions and Project Items

See Also

Solution Explorer
Projects (SQL Server Management Studio)
Solution Explorer Source Control

Feedback

Was this page helpful? Yes No

Associate a Query with a Connection in a Project

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

If a query was created without a connection, or if a query is moved from one project to another it will not be associated with a connection in the current project.

To associate a query with a connection in a project

1. If the query is open in query editor, right-click a blank area of query editor, point to **Connection**, and then click **Connect**. If the query is not open, double-click the query in Solution Explorer to connect the query.

2. In the **Connect to Database Engine** dialog box, provide the connection information. If the connection information matches an existing connection, the query will be associated with that connection.

See Also

Solution Explorer
Change the Connection Associated with a Query
View or Change the Properties of a Connection in a Project

Feedback

Was this page helpful? Yes No

Projects (SQL Server Management Studio)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

A Management Studio project is a collection of logically related scripts and files that can be saved together for database administration and development.

Script Project Overview

SQL Server script projects are displayed in the Solution Explorer component of Management Studio. A script project can contain zero or more project files. You can add a project to a solution or combine more than one project within a solution.

Projects can include the following:

- **Connections**. A connection, as persisted within a project, will contain login information, server name, default database, preferred protocol, authentication type, and connection properties. Connection information may optionally be stored with a script (see below).
- **SQLScripts**. Frequently used SQL scripts for the user. Double-clicking a .sql file within the project will cause the SQL Editor to open the selected script.
- **MDX, DMX, and XMLAScripts**. Frequently used MDX scripts for the user. Double-clicking an .mdx file within the project will cause the Editor to open the selected script.
- **Misc**. This folder can be used for files that do not neatly fit into any of the other default node types, such as a text file that contains the project objectives.

Projects can also be integrated into a source code control system.

Connecting to an Instance of SQL Server from a Script Project

A script project may contain connections to an instance of SQL Server. You can connect to an instance of SQL Server in a project by clicking the connection. This will open a SQL

Script window that is connected to the instance of SQL Server defined in the connection you selected. If you open a SQL Server or MDX script with a connection that uses SQL Server authentication, you will be prompted for the password using the **Connect to SQL Server** dialog box after the Editor has been opened and the script loaded.

The connection will be closed after the corresponding window is closed.

To modify information about a connection, use the properties window in Management Studio.

Project Tasks

Expand table

Task Description	Topic
Describes how to create a new project in a solution.	Create a Project
Describes how to add an existing project to a solution.	Add an Existing Project to a Solution
Describes how to change the default location at which project files are saved.	Change the Default Location for Projects
Describes how to view the current properties for a project.	View Project Properties
Describes how to add new items, such as connections or script files, to a project.	Add New Items to a Project
Describes how to establish the connection information for a query.	Associate a Query with a Connection in a Project
Describes how to change the connection information for a query.	Change the Connection Associated with a Query
Describes how to change connection properties.	View or Change the Properties of a Connection in a Project

See Also

Solution Explorer
Solutions (SQL Server Management Studio)
Solution Explorer Source Control

Feedback

Was this page helpful? Yes No

Move Items in a Solution

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Project items in SQL Server Management Studio projects are Queries, Connections, and Miscellaneous files. You can move Queries and Miscellaneous Files between projects in Solution Explorer, but Connections cannot be moved.

To move items in Solution Explorer

1. In Solution Explorer, select the item you want to move.
2. On the **Edit** menu, click **Cut**.
3. In Solution Explorer, select the destination.
4. On the **Edit** menu, click **Paste**.

You can move items by dragging Query and Miscellaneous files within Solution Explorer. Dragging allows you to see the outcome of the drag operation. Moving queries from one project type to another may cause queries to be considered as miscellaneous files in the target project.

> **Note**
>
> Moving a connected query does not move the connection to the target project. The query will lose its connection after it is moved to the target project.

See Also

Solution Explorer
Copy Items in a Solution

Feedback

Was this page helpful? Yes No

Manage Files with Encoding

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

To facilitate the display of your code in a particular language and on a particular platform, you can associate a particular character encoding with a file.

Opening Files

You can choose the editor you want to use to edit the file.

To open a file with a specific editor

1. On the **File** menu, point to **Open**, and then click **File**.
2. In the **Open File** dialog box, select the file name.
3. Click the arrow next to the **Open** button, and then click**Open With** from the menu that appears.
4. In the **Select a program to open** list, select an editor, and then click **Open**. To open the file with a particular encoding, select an editor with encoding support, such as SQL Query Editor with Encoding or XML Editor with Encoding.

Saving Files

You can also save your code with a Unicode encoding or a different code page to support various languages, such as Western European or Eastern European. You can associate a particular character encoding with a file to facilitate the display of your code in that language, as well as a line-ending type to support a particular operating system. Also, some characters, when used in file names, cannot be saved unless they are saved with Unicode encoding.

To save a file with a different encoding or line ending type

1. On the **File** menu, click **Save <filename> As**.
2. In the **Save File As** dialog, expand the **Save** button, and then click **Save with Encoding**.

3. In the **Advanced Save Options** dialog box, select the encoding you want from the **Encoding** list.

4. From the **Line Endings**list, select the type of line ending you want.

> **Note**
>
> If you save your file your file with Unicode encoding, the file should be checked into Microsoft Visual SourceSafe as a binary file because Visual SourceSafe does not support merging, comparing, and showing differences between files that are saved as Unicode.

If you are using Visual SourceSafe to store files with ANSI, UTF8, or Unicode, be aware of the following limitations for each:

- ANSI files allow only characters that are supported in the current code page, which limits international use.
- Unicode files cannot use shared checkout, difference checking, or merging functionality because they are handled as binary files. You can use this format in international files.
- UTF8 files do not work safely with Visual SourceSafe because changes that cause problems for UTF8 file editors are made during check in, check out, difference checking, and merging.

See Also

Files That Manage Solutions and Projects
Associating File Extensions to a Code Editor

Feedback

Was this page helpful? Yes No

Change the Connection Associated with a Query

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

A query can be executed against several different servers or databases by changing the connection used by the query.

To change the connection associated with a query

1. In Query Editor, right-click a blank area of Query Editor, point to **Connection**, and then click **Change Connection**.
2. In the **Connect to Database Engine** dialog box, provide the new connection information, and then click **Connect**.

See Also

Solution Explorer
Associate a Query with a Connection in a Project
View or Change the Properties of a Connection in a Project

Feedback

Was this page helpful? Yes No

Add an Existing Project to a Solution

Article • 02/19/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can add one or more projects to an existing solution. A project can be associated with more than one solution.

To add an existing project to a solution

1. In Solution Explorer, select the solution.
2. On the **File** menu, point to **Add**, and select **Existing Project**.
3. In the **Add Existing Project** dialog box, locate the project you want to add, select the project file, and then select **Open**.

   The project is added to the selected solution.

> **Note**
>
> Before adding an existing project from shared directories in a multideveloper environment, you should attempt to ensure that your changes do not adversely affect or overwrite another developer's changes.

Related content

- Projects (SQL Server Management Studio)
- Solutions (SQL Server Management Studio)

Feedback

Was this page helpful? Yes No

Create a Project

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can create one or more projects within an existing solution.

Create a new project and add it to a solution

1. In Solution Explorer, select the solution.
2. On the **File** menu, point to **Add**, and click **New Project**.
3. In the **New Project** dialog box, click a type of project.

   Templates
   In the **Templates** box, select a template. A brief description of the selected project template appears beneath the **Templates** box.

   Name
   Enter the name of the script project you want to create. A folder with the same name as the project is also created in the location displayed in the **Location** field. For some projects, SQL Server Management Studio creates source and other supporting files and adds them to the new project folder.

   > **Note**
   >
   > For some project types, the **Name** text box is unavailable because specifying the location sets the name. For example, Web applications and Web services are located on a Web server and derive their name from the virtual directory specified on that server.

   Names cannot contain the following characters:

   - Pound (#)
   - Percent (%)
   - Ampersand (&)
   - Asterisk (*)

- Vertical bar (|)
- Backslash (\)
- Colon (:)
- Quotation mark (")
- Less than (<)
- Greater than (>)
- Question mark (?)
- Forward slash (/)
- Leading or trailing spaces (' ')
- Names reserved for Microsoft Windows or MS-DOS, such as ("nul", "aux", "con", "com1", "lpt1", and so on)

Location
Enter the location where you want to create your project, or choose one from the list.

Browse
Displays the **Project Location** dialog box, allowing you to navigate to a new directory in which to save the project.

Solution
Select **Create new Solution** to create a solution in Solution Explorer. Select **Add to Solution** to add the new project to the solution currently open in Solution Explorer.

Create directory for Solution
Select to enable the **(Solution) Name** text box. This option creates a new directory of the name you choose for your project and solution.

Solution Name
Enter the name of the new solution in which you want your project to be created. By default, this field uses the same name as entered in the **Name** field.

Note To access this option, you must select both the **Create New Solution** in the **Solution** and the **Create directory for Solution** check boxes. Some project templates do not support this option, such as Web applications.

Add to Source Control
When this check box is selected, your source control application opens when you

click **OK**. Complete any information required by your source control application to continue. You must have a source control client application installed to use this option.

4. Click **OK**.

You can set a name for the script project, but the folder names are established by Management Studio and cannot be changed. You can configure the drive and path specification for the common set of folders by using the **Add New Project** dialog box. Right-click the solution icon in **Solution Explorer**, and then click **Add**. The default location for script project folders is: C:\Documents and Settings*username*\My Documents\SQL Server Management Studio\Projects\.

See Also

Solution Explorer
Add an Existing Project to a Solution
Add New Items to a Project
Add Existing Items to a Project
Change the Default Location for Projects
Remove or Delete an Item or Project
Delete a Solution

Feedback

Was this page helpful? Yes No

Add New Items to a Project

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Add new items to a project to extend application functionality. A new item can be a query or a connection. SQL Server Management Studio has two project types: SQL Server Script Project, and Analysis Services Script Project. The project type determines the items that you can add to the project. For example, you can add a Transact-SQL query (a file with a .sql extension) to a SQL Server Script project, but you cannot add it to an Analysis Services Script Project.

SQL Server Management Studio does not allow you to create folders within projects. To organize your work, create multiple projects within the solution.

To add a new query to an existing project

1. In Solution Explorer, select a target project.
2. On the **Project** menu, click **Add New Item**.
3. In the **Add New Item** dialog box, select a category in the left pane.
4. Select a query template in the right pane, and then click **Add**. The new query is added in the **Queries** folder of the project.
5. In the **Connect to Database Engine** dialog box, specify a connection for the new query, and then click **Connect**. You can click **Cancel** on the connection dialog if you do not want to associate a connection to the new query.
6. Rename the query in Solution Explorer if you wish.

To add a new connection to an existing project

1. In Solution Explorer, select a target project.
2. On the **Project** menu, click **Add New Item**.
3. Select **Connection** in the left pane.
4. Select **New Connection** in the right pane, and then click **Add**.

5. In the **Connect to Database Engine** dialog box, specify a connection for the new query, and then click **Connect**. The new connection gets added in the **Connections** folder of the project.

See Also

Solution Explorer
Associating File Extensions to a Code Editor
Add Existing Items to a Project
Remove or Delete an Item or Project

Feedback

Was this page helpful? Yes No

Delete a Solution

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Deleting a solution permanently erases files from physical storage as well as the file items in the solution.

> **Note**
>
> In contrast to projects and items, a solution can only be deleted permanently, not removed temporarily.

To delete a solution

1. In Solution Explorer, select the solution you want to delete.
2. On the **File** menu, click **Close Solution**.
3. In Microsoft Windows Explorer, locate and select the files associated with the solution you want to delete.
4. On the **File** menu, click **Delete**.

See Also

Solution Explorer
Create a New Solution
Create a Project
Remove or Delete an Item or Project

Feedback

Was this page helpful? Yes No

Open an Existing Solution

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can have only one solution at a time open in Solution Explorer.

To open an existing solution

1. On the **File** menu, point to **Open**, and then click **Project/Solution**.
2. In the **Open Project** dialog box, navigate to the solution you want, click the solution, and then click **Open**.

See Also

Solution Explorer
Create a New Solution

Feedback

Was this page helpful? Yes No

View Project Properties

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can view properties associated with your projects in the Properties window.

To view project properties

1. In Solution Explorer, select a project.
2. If the Properties window is not open, on the **View** menu, click **Properties Window**.
3. View the properties of the project in the Properties window.

Project properties, such as the date created, are read-only and cannot be changed from the Properties window.

See Also

Solution Explorer

Feedback

Was this page helpful? Yes No

Remove or Delete an Item or Project

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Project items in SQL Server Management Studio projects are Queries, Connections, and Miscellaneous files. You can remove project queries and miscellaneous files from your solution without erasing the files from storage. Remove a project or item when it is not useful in the current solution but you want to include it in another solution.

To remove a project item

1. In Solution Explorer, select the project item you want to remove.
2. On the **Edit** menu, click **Remove**.
3. On the confirmation dialog, click **Remove** to remove the item from the project.

A removed item still exists on the file system. Therefore, you can add a removed item to its original solution or to another solution.

To remove a project

1. In Solution Explorer, select the project you want to remove.
2. On the **Edit** menu, click **Remove**.
3. On the confirmation dialog, click **OK**, to remove the project from the solution.

You can delete a project permanently, but you first need to remove any references to the project from SQL Server Management Studio solutions, and then use Microsoft Windows Explorer to permanently delete the associated files from storage.

To delete a project

1. In Solution Explorer, remove the project you want to delete from the solution.
2. In Windows Explorer, locate and select the files associated with the project or item you want to delete.
3. On the **File** menu, click **Delete**.

See Also

Solution Explorer
Add New Items to a Project
Add Existing Items to a Project

Feedback

Was this page helpful? Yes No

Miscellaneous Files

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Files that are external to any project are called *miscellaneous files*. When you have a solution open, you can open and modify miscellaneous files related to the project. A file is classified as a miscellaneous file if the file extension is not associated with the project code editor. For example, in a Microsoft SQL Server Scripts Project, files with the extension .txt or .mdx will be treated as miscellaneous files. In an MDX project, files with the extension of .txt or .sql will be treated as miscellaneous files. To associate a file extension with a code editor, see How to: Associate File Extensions to a Code Editor.

There are a number of reasons why it is useful to be able to add miscellaneous files to your project. You might have a file that is not necessarily a recognized script but integral to the solution's development. Common examples include development notes or instructions, data files, and code clips.

Miscellaneous files provide flexibility. For example, suppose you have a SQL Server Scripts Project that has several scripts for creation of tables and stored procedures in your database. You also have several data files for the tables with file extensions .BCP and execution instructions in a README.TXT file. You can attach the data and the README files as miscellaneous files to the project to take advantage of source control and other features of the project system.

SQL Server Management Studio menus and toolbars change according to the format of the file you open. When you open a text file, for example, the Text Editor toolbar appears. If you open an XML Schema file, the XML Schema toolbar appears. While editing your XML Schema, the Text Editor toolbar is unavailable. When you switch between a project file and a miscellaneous file, all project-related commands and toolbars are replaced by those relevant to the miscellaneous file.

See Also

Files That Manage Solutions and Projects
Solutions (SQL Server Management Studio)
Projects (SQL Server Management Studio)

Feedback

Was this page helpful? Yes No

Change the Default Location for Projects

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The default location for files and projects is the SQL Server Management Studio Projects folder in your My Documents folder.

To change the default location for projects

1. Change the default location by opening or saving a file or project from another location. Microsoft SQL Server Management Studio stores the location of the previously opened project and uses it for the next opening or saving of a project.

See Also

Solution Explorer

Feedback

Was this page helpful? Yes No

Rename Solutions and Project Items

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can rename a solution and its project items in Solution Explorer.

> **Note**
>
> Projects and connections cannot be renamed.

To rename a solution or item

1. In Solution Explorer, select the solution or the project item that you want to rename.
2. On the **View** menu, click **Properties Window**.
3. In the Properties window, edit the **(Name)** property.

See Also

Solution Explorer

Feedback

Was this page helpful? Yes No

Close a Solution

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can close a solution and remove it from Solution Explorer. Closing a solution makes it unavailable for use by Solution Explorer.

> **Note**
>
> Closing Solution Explorer does not automatically close the solution. It only hides the Solution Explorer window.

To close a solution

1. On the **File** menu, click **Close Solution**.

See Also

Solution Explorer

Feedback

Was this page helpful? Yes No

View or Change the Properties of a Connection in a Project

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

The properties of a connection saved in a project can be viewed and some can be modified.

To change the properties of a connection in a project

1. In Solution Explorer, right-click the connection, and then click **Properties Window**.
2. View the connection properties in the Properties window. You can change the **Execution Timeout**, **Database**, and **Login Timeout** properties.

See Also

Solution Explorer
Associate a Query with a Connection in a Project
Change the Connection Associated with a Query

Feedback

Was this page helpful? Yes No

Add Existing Items to a Project

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Add new items to a project to extend application functionality. An existing item can be a query or a miscellaneous file. SQL Server Management Studio has two project types: SQL Server Script Project, and Analysis Services Script Project. The project type determines the query files that you can add to the project. For example, you can add a Transact-SQL query (a file with a .sql extension) to a SQL Server Script project, but you cannot add it to an Analysis Services Script Project. To associate additional file extensions to a project type, see How to: Associate File Extensions to a Code Editor.

To add an existing query or a miscellaneous file to a project

1. In Solution Explorer, select a target project.
2. On the **Project** menu, click **Add Existing Item**.

Look in
Locate the files or folders to add to your project from this list. For XML Web services and ASP.NET Web applications, the files are located on the Web server.

Desktop
Displays the files and folders located on the desktop.

My Projects
Displays the files and folders at the default **My Projects** location.

My Computer
Displays the contents of your **My Computer** folder.

File name
Use this option to filter the files and folders that are displayed. Enter the full or partial file name on which to filter; use an asterisk (`*`) as a wildcard.

> **Note**
>
> Navigate to Web and network locations by entering the URL or network path in the **File name** box. For example, `https://mywebsite` displays the files

available at the mywebsite Web location, and **\\myserver\myshare** displays the files available at the myshare location on myserver.

Files of type
Use this option to filter files based on file extension. Each product lists default filters of the most common file types.

Add
Use this drop-down button to add the item to the project and open the item in its default editor.

- **Add**

  Copies the existing item to your project folder on disk and adds the item under the selected project in Solution Explorer. Any changes made to the item are not reflected in the item at the original location. This is the default editor for the file type.

- **Add As Link**

  Adds the selected file to the project and opens it with the default editor for the file type. This option opens the originally selected file and does not copy the file to the project folder.

3. If you are adding query files, the connection dialog will prompt you to specify a connection for the query. You can click **Cancel** on the connection dialog if you do not want to associate a connection to the query.

4. The file will be added to the **Queries** or **Miscellaneous Files** folder of the project.

See Also

Solution Explorer
Add New Items to a Project
Remove or Delete an Item or Project

Feedback

Was this page helpful? Yes No

Create a New Solution

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

When you create a project, SQL Server Management Studio creates a solution to contain it. You can only open one solution at a time. SQL Server Management Studio does not allow you to create folders within projects. To organize your work, create multiple projects.

To access Solution Explorer, click **Solution Explorer** on the **View** menu.

To create a new solution and project

1. On the **File** menu, point to **New**, and then click **Project**.
2. In the **New Project** dialog box, click a type of project.
3. Optionally modify the project name in the **Name** text box.
4. Optionally modify the path for the solution in the **Location** text box.
5. Optionally modify the solution name in the **Solution Name** text box.
6. Click **OK**.

See Also

Solution Explorer
Open Solutions from Source Control

Feedback

Was this page helpful? Yes No

Template Explorer

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

SQL Server provides a variety of templates. Templates are boilerplate files containing SQL scripts that help you create objects in a database. The first time the template explorer is opened, a copy of the templates are placed in the user's folder in C:\Users, under AppData\Roaming\Microsoft\SQL Server Management Studio\130\Templates.

You can browse the available templates in Template Explorer, then open a template to incorporate the code into a code editor window. You can also create custom templates.

Benefits of Templates

Templates are available for solutions, projects, and various types of code editors. Templates are available to create objects like databases, tables, views, indexes, stored procedures, triggers, statistics, and functions. In addition, there are templates that help you to manage your server by creating extended properties, linked servers, logins, roles, users, and templates for Analysis Services.

The template scripts provided with SQL Server Management Studio contain parameters to help you customize the code. When you open a template, Use the **Specify Values for Template Parameters** dialog box to insert values into the script.

Create custom templates for tasks you perform frequently. Organize your custom scripts into the existing folders or create a new folder structure.

The Database Engine Query editor also supports code snippets, which can be inserted at specific locations in a script by right-clicking at that location.

Related Tasks

Use the following topics to get started with templates

Expand table

Description	Topic
Describes how to incorporate the code from a template into a code	Open a Template

Description	Topic
editor window.	
Describes how to replace template parameter values after opening a template in a code editor.	Replace Template Parameters

Feedback

Was this page helpful? Yes No

Open a Template

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

You can open a template from Template Explorer.

To open a template from Template Explorer

You can open templates from the Template Explorer.

1. To open Template Explorer, on the **View** menu, click **Template Explorer**.
2. In the list of template categories, expand the category that contains the template you want to open.
3. There are three ways to open the template:
   a. Double-click the template to open it in a code editor window.
   b. Right-click the template and select **Open** to open it in a code editor window.
   c. Drag the template into a code editor window to add the template code to the contents of the editor window.

After the template is open, use the **Specify Values for Template Parameters** dialog box from the **Query** menu to replace the template parameters with your values.

If opening a template launches a new editor window, the window will open with the credentials of the current active connection. For example, if you are focused on an instance of the Database Engine in Object Explorer when you open the CREATE DATABASE template, a new editor window will be opened using a connection to that instance. If there is no active connection, Management Studio will present a login dialog.

See Also

Template Explorer
Replace Template Parameters

Feedback

Was this page helpful? Yes No

Replace Template Parameters

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

Templates contain parameters that can be replaced by implementation-specific values each time the template is used. After opening a template in a code editor, you can replace the parameters with values relevant to your implementation.

Before You Begin

The **Specify Values for Template Parameters** dialog is a grid with three columns. The **Parameter** and **Type** columns are read-only and cannot be changed. Review the contents of the **Value** column, and change any of the defaults to values appropriate for your implementation.

To use this dialog box, you must have parameters in your script enclosed in angle brackets (`< >`) in the format `<`*parameter_name*`,` *data_type*`,` *default_value*`>`.

Replace Template Parameters

After opening the template in a code editor window:

1. On the **Query** menu, click **Specify Values for Template Parameters**.
2. In the **Specify Values for Template Parameters** dialog box, the **Values** column contains a suggested value for each parameter. Accept the value or replace it with a new value.
3. Click **OK** to close the **Specify Values for Template Parameters** dialog box and modify the script in the query editor.

See Also

Template Explorer
Open a Template

Feedback

Was this page helpful? Yes No

Visual Database Tools

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

SQL Server Management Studio includes visual designers for building Transact-SQL queries, tables, and diagramming databases.

Related Tasks

Use the following tasks to get started with Visual Database Tools:

Expand table

Description	Topic
Describes the database diagram tool.	Design Database Diagrams
Describes the visual table design tool.	Design Tables
Describes the visual query designer.	Design Queries and Views How-to articles

Feedback

Was this page helpful? Yes No

Reconcile Changes Made by Multiple Users (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

In a multiuser environment, changes can be made on the same object by multiple users at once. This can happen when you're working on the structure of the object in the Table or Database Diagram designers or it can happen to values in the results returned in the Query and View designer's Results pane. This can cause conflicts that you'll want to resolve.

Conflicts in the Table or Database Diagram Designers

For example, another user might delete or rename a table while you are working with the same or a related table in Table Designer. When you attempt to save your table, the Database Changes Detected dialog box notifies you that the database has been updated since you opened the table.

This dialog box also displays a list of database objects that will be affected as a result of saving your table. At this point, you can take one of these actions:

- Choose **Yes** to save your table and update the database with all the changes in the list.

  This action could affect tables that share the same database objects. For example, suppose you edit the `au_id` column in the `titleauthors` table while another user is working on the `authors` table which is related to the `titleauthors` table by the `au_id` column. Saving your table will affect the other user's table. Similarly, suppose that another user defined a check constraint for the `qty` column in the `sales` table. If you delete the `qty` column and save the `sales` table, the other user's check constraint will be affected.

- Choose **No** to cancel the save action.

  You can then close the table without saving it. When you reopen the table it will match what is in the database.

- Choose **Save Text File** to save a list of the changes.

You can save the list of database changes shown in the **Database Changes Detected** dialog box to a text file so that you can investigate the cause of other users' changes. For example, if another user edited a table that you marked for deletion, you may want to research whether the table should be deleted before updating the database.

Conflicts in the Query and View Designer

If you run a query or return the results of a view, the data is shown in the Results Pane. Multiple users can work on the same set of data at the same time, which can cause conflicts.

For example, lets say you and a colleague each run a query to show all the data in the `titleauthors` table. Your colleague changes the first name in the first record returned from Barb to Barbara. At this point the database has Barbara in that field, while your result set still shows Barb. Now you type in Barbara and click off of the row. You will receive a message asking you how you want to resolve the conflict.

- Click **Yes** to update the database with your changes.

  This will override your colleague's changes.

- Click **No** to have your result set updated to what's currently in the database.

  This will override your changes with those of your colleague's.

- Click **Cancel** to continue to edit without resolving the conflict.

  In this case you will not be able to commit your changes to the database.

Feedback

Was this page helpful? Yes No

Create Self-Joins Automatically (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

If a table has a reflexive relationship in the database, you can join it to itself automatically.

To create a self-join automatically

1. Add to the Diagram pane the table you want to work with.
2. Add the same table again, so that the Diagram pane shows the same table twice within the Diagram pane.

   The Query and View Designer assigns an alias to the second instance by adding a sequential number to the table name. In addition, the Query and View Designer creates a join line between the two rectangles representing the two different ways the table participates in the query.

See Also

Query with Joins (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Object Already Exists

Article • 02/13/2025

Applies to: SQL Server

The object you are trying to save has the same name as an object already in the database. Choose another name and proceed.

Can also indicate the column name already exists.

Feedback

Was this page helpful? Yes No

Summarize or Aggregate Values for All Rows in a Table (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

Aggregate function

Using an aggregate function, you can create a summary for all the values in a table. For example, you can create a query such as the following to display the total price for all books in the `titles` table:

```
SELECT SUM(price)
FROM titles
```

Create multiple aggregations in the same query by using aggregate functions with more than one column. For example, you can create a query that calculates the total of the `price` column and the average of the `discount` column.

You can aggregate the same column in different ways (such as totaling, counting, and averaging) in the same query. For example, the following query averages and summarizes the `price` column from the `titles` table:

```
SELECT AVG(price), SUM(price)
FROM titles
```

If you add a search condition, you can aggregate the subset of rows that meet that condition.

> **Note**
>
> You can also count all the rows in the table or the ones that meet a specific condition. For details, see Count Rows in a Table (Visual Database Tools).

When you create a single aggregation value for all rows in a table, you display only the aggregate values themselves. For example, if you are totaling the value of the `price`

column of the `titles` table, you would not also display individual titles, publisher names, and so on.

! If you are subtotaling - that is, creating groups - you can display column values for each group. For details, see Group Rows in Query Results (Visual Database Tools).

Aggregate values for all rows

1. Be sure the table you want to aggregate is already present in the Diagram pane.
2. Right-click the background of the Diagram pane, then choose **Group By** from the shortcut menu. The Query and View Designer adds a **Group By** column to the grid in the Criteria pane.
3. Add the column you want to aggregate to the Criteria pane. Be sure that the column is marked for output.

   The Query and View Designer automatically assigns a column alias to the column you are summarizing. You can replace this alias with a more meaningful one. For details, see Create Column Aliases (Visual Database Tools).
4. In the **Group By** grid column, select the appropriate aggregate function, such as: **Sum**, **Avg**, **Min**, **Max**, **Count**. If you want to aggregate only unique rows in the result set, choose an aggregate function with the DISTINCT options, such as **Min Distinct**. Do not choose **Group By**, **Expression**, or **Where**, because those options do not apply when you are aggregating all rows.

   The Query and View Designer replaces the column name in the statement in the SQL pane with the aggregate function that you specify. For example, the SQL statement might look like this:

   ```
   SELECT SUM(price)
   FROM titles
   ```
5. If you want to create more than one aggregation in the query, repeat steps 3 and 4.

   When you add another column to the query output list or order by list, the Query and View Designer automatically fills the term **Group By** into the **Group By** column of the grid. Select the appropriate aggregate function.
6. Add search conditions, if any, to specify the subset of rows you want to summarize.

When you execute the query, the Results pane displays the aggregations that you specified.

> **Note**
>
> The Query and View Designer maintains aggregate functions as part of the SQL statement in the SQL pane until you explicitly turn off Group By mode. Therefore, if you modify your query by changing its type or by changing which tables or table-valued objects are present in the Diagram pane, the resulting query might include invalid aggregate functions.

See Also

Sort and Group Query Results (Visual Database Tools)
Summarize Query Results (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Join Tables Automatically (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

When you add two or more tables to a query, the Query and View Designer attempts to determine if they are related. If they are, the Query and View Designer automatically puts join lines between the rectangles representing the tables or table-structured objects.

The Query and View Designer will recognize tables as joined if:

- The database contains information that specifies that the tables are related.
- If two columns, one in each table, have the same name and data type. The column must be a primary key in at least one of the tables. For example, if you add `employee` and `jobs` tables, if the `job_id` column is the primary key in the `jobs` table, and if each table has a column called `job_id` with the same data type, the Query and View Designer will automatically join the tables.

  > **Note**
  >
  > The Query and View Designer will create only one join based on columns with the same name and data type. If more than one join is possible, the Query and View Designer stops after creating a join based on the first set of matching columns that it finds.

- The Query and View Designer detects that a search condition (a WHERE clause) is actually a join condition. For example, you might add the tables `employee` and `jobs`, then create a search condition that searches for the same value in the `job_id` column of both tables. When you do, the Query and View Designer detects that the search condition results in a join, and then creates a join condition based on the search condition.

If the Query and View Designer has created a join that is not suitable to your query, you can modify the join or remove it. For details, see Modify Join Operators (Visual Database Tools) and Remove Joins (Visual Database Tools).

If the Query and View Designer does not automatically join the tables in your query, you can create a join yourself. For details, see Join Tables Manually (Visual Database Tools).

See Also

How the Query and View Designer Represents Joins (Visual Database Tools)
Design Queries and Views How-to Topics (Visual Database Tools)
Query with Joins (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Work with Tables in Database Diagram (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server · Azure SQL Database · Azure SQL Managed Instance · Analytics Platform System (PDW)

You can modify and create database tables in either Table Designer or Database Diagram Designer.

> **Note**
>
> If the table is published for replication, you must make schema changes using the Transact-SQL statement **ALTER TABLE** or SQL Server Management Objects (SMO). When schema changes are made using the Table Designer or the Database Diagram Designer, it attempts to drop and recreate the table. You cannot drop published objects, therefore the schema change will fail.

In This Section

Add Tables to Diagrams (Visual Database Tools)

Add Related Tables to Diagrams (Visual Database Tools)

Save Selected Tables on a Diagram (Visual Database Tools)

Copy Tables from One Database Diagrams to Another (Visual Database Tools)

Remove Tables from Database Diagrams (Visual Database Tools)

Map Many-to-Many Relationships (Visual Database Tools)

Draw Reflexive Relationships (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Reconcile a Database Diagram with a Modified Database (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

You save your database diagram when you are ready to update the database to match your diagram. However, if other users have updated the database since you opened your diagram, their changes might affect your diagram and vice versa.

Saving your diagram will reconcile the database with your diagram by overwriting other users' changes so that the database will match your diagram.

To update a database to match your diagram

1. Save your database diagram.

   If you have not previously saved your diagram, type a name for the diagram in the **Save New Database Diagram** dialog box and choose **OK**.

2. The **Save** dialog box lists the tables that will be affected when you save your diagram. Choose **Yes** to continue.

3. The **Database Changes Detected** dialog box lists the objects that were modified and will be changed to match your diagram. Choose **Yes** to save the diagram and accept the list of changes.

   > **Note**
   >
   > If your diagram contains tables and columns that were deleted in the database, only their definitions are recreated in the database when you save your diagram. This process does not restore any data that existed in these objects before their deletion.

To update your diagram to match a modified database

1. Close your diagram without saving changes.

2. Right-click the diagram in Object Explorer.

3. From the shortcut menu click **Refresh**.

4. Reopen the diagram.

See Also

Work with Database Diagrams (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Autosize Selected Tables in Diagrams (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

You can quickly fit tables in your diagram to the smallest size that shows all fields.

To automatically size selected tables in diagrams

1. Select the table(s) you want to resize.
2. Right-click a selected table and from the shortcut menu choose **Autosize Selected Tables**.

See Also

Work with Database Diagrams (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Table-Valued Object Properties (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

These properties appear in the Properties window when you select a table-valued object in **Query and View Designer**. The table-valued object could be a view, synonym, derived table, or table-valued function. Unless otherwise noted, these properties are read-only in the **Properties** window.

> **Note**
>
> The properties in this topic are ordered by category rather than alphabet.

> **Note**
>
> The dialog boxes and menu commands you see might differ from those described in Help depending on your active settings or edition. To change your settings, choose **Import and Export Settings** on the **Tools** menu.

Identity Category
Expands to show the **Name** and **TVO Type** properties.

Name
Shows the name of the selected table-valued object.

TVO Type
Shows which type of table-valued object. It can be either a base table, view, table-valued function, or a derived table.

Query DesignerCategory
Expands to show properties for **Alias**, **Column List**, **Full Name**, and **Parameter List**.

Alias
Shows the alias for the selected table-valued object. To add or change an alias, type it into the field.

Column List
Shows the columns included in the selected table-valued object. To see them in a

separate window, click Column List and then click the ellipses (...) to the right of the property.

Full Name
Shows the name of the selected table-valued object, including additional information such as the schema or data source of the object.

Parameter List
Shows the parameters defined for selected table-valued function. To define a value for the parameters, click Parameter List and then click the ellipses (...) to the right of the property. In the Function Parameters dialog box, type in values. This property is only available when a table-valued function is selected.

Feedback

Was this page helpful? Yes No

View Join Properties (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can see a collection of properties for a view in the Properties window.

To view join properties

1. Right-click the join in the **Diagram** pane.
2. From the shortcut menu click **Properties**.

   The join's properties show in the Properties window.

See Also

Query with Joins (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Column Properties (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

There are two sets of properties for columns: a full set that you can see in the **Column Properties** tab within Table Designer (available only for Microsoft SQL Server databases) and a subset you can see in the Properties window using Server Explorer.

> **Note**
>
> The properties in this topic are ordered by category rather than alphabet.

> **Note**
>
> The dialog boxes and menu commands you see might differ from those described in Help depending on your active settings or edition. To change your settings, choose **Import and Export Settings** on the **Tools** menu.

Properties Window

These properties appear in the Properties window when you select a column in Server Explorer.

> **Note**
>
> These properties, accessed using Server Explorer, are read-only. To edit column properties for SQL Server databases, select the column in Table Designer. Those properties are described later in this topic.

Identity Category
Expands to show the **Name** and **Database** properties.

Name
Shows the name of the column.

Database
Shows the name of the data source for the selected column. (Applies only to OLE DB.)

Misc Category
Expands to show the remaining properties.

Data Type
Shows the data type of the selected column. For more information, see Data Types (Transact-SQL).

Identity Increment
Shows the increment that will be added to the **Identity Seed** for each subsequent row of the identity column. (Applies only to SQL Server.)

Identity Seed
Shows the seed value assigned to the first row in the table for the identity column. (Applies only to SQL Server.)

Is Identity
Shows whether the selected column is the identity column for the table. (Applies only to SQL Server.)

Length
Shows the number of characters allowed for character-based data types.

Nullable
Shows whether or not the column allows null values.

Precision
Shows the maximum number of digits allowed for numeric data types. This property shows **0** for nonnumeric data types.

Scale
Shows the maximum number of digits that can appear to the right of the decimal point for numeric data types. This value must be less than or equal to the precision. This property shows **0** for nonnumeric data types.

Column Properties Tab

To access these properties, in Server Explorer right-click the table to which the column belongs, choose **Open Table Definition**, and select the row in the table grid in Table Designer.

> **Note**
>
> These properties apply only to SQL Server.

General Category
Expands to show **Name**, **Allow Nulls**, **Data Type**, **Default Value or Binding**, **Length**, **Precision**, and **Scale**.

Name
Displays the name of the column. To edit the name, type in the text box.

> **Caution**
>
> If existing queries, views, user-defined functions, stored procedures, or programs refer to the column, the name modification will make these objects invalid.

Allow Nulls
Shows whether or not the column's data type allows null values.

Data Type
Shows the data type for the selected column. To edit this property, click its value, expand the drop-down list, and choose another value. For more information, see Data Types (Transact-SQL).

Default Value or Binding
Shows the default for this column when no value is specified for this column. The drop-down list contains all global defaults defined in the data source. To bind the column to a global default, select from the drop-down list. Alternatively, to create a default constraint for the column, type the default value directly as text.

Length
Shows the number of characters allowed for character-based data types. This property is only available for character-based data types.

Precision
Shows the maximum number of digits allowed for numeric data types. This property shows **0** for nonnumeric data types. This property is only available for numeric data types.

Scale
Shows the maximum number of digits that can appear to the right of the decimal point for numeric data types. This value must be less than or equal to the precision. This

property shows **0** for nonnumeric data types. This property is only available for numeric data types.

Table Designer Category
Expands to show the remaining properties.

Collation
Shows the collation setting for the selected column. To change this setting, click **Collation** and then click the ellipses **(...)** to the right of the value.

Computed Column Specification Category
Expands to show properties for **Formula** and **Is Persisted**. If the column is computed, the formula will also be displayed. To edit the formula, expand this category and edit it in the **Formula** property.

Formula
Shows the formula that the selected column uses if it is a computed column. In this field you can enter or change a formula.

Is Persisted
Allows you to save the computed column with the data source. A persisted computed column can be indexed.

Condensed Data Type
Displays information about the field's data type, in the same format as the SQL CREATE TABLE statement. For example, a field containing a variable-length string with a maximum length of 20 characters would be represented as "varchar(20)." To change this property, type the value directly.

Description
Shows the description of the column. To see the full description or to edit it, click Description, and then click the ellipses **(...)** to the right of the property.

Full-text Specification Category
Expands to show properties specific to full-text columns.

Is Full-text Indexed
Indicates whether this column is full-text indexed. This property can be set to **Yes** only if the data type for this column is full-text searchable and if the table to which this column belongs has a full-text index specified for it. To change this value, click it, expand the drop-down list, and choose a new value.

Full-text type column
Shows which column is used to define the document type of a column of type image.

The image data type can be used to store documents ranging from .doc files to xml files.

Language
Indicates the language used to index the column.

Statistical Semantics
Select whether to enable statistical semantic indexing for the selected column. For more information, see Semantic Search placeholder.

If you select a **Language** prior to selecting **Statistical Semantics**, and the selected language does not have an associated Semantic Language Model, then the **Statistical Semantics** option is set to **No** and cannot be modified. If you select **Yes** for the **Statistical Semantics** option prior to selecting a **Language**, then the languages available in the **Language** column will be restricted to those for which there is Semantic Language Model support.

Has Non-SQL Server Subscriber
Shows whether the column has a non-Microsoft SQL Server subscriber.

Identity Specification Category
Expands to show properties for **Is Identity**, **Identity Increment**, and **Identity Seed**.

Is Identity
Shows whether the selected column is the identity column for the table. To change the property, open the table in Table Designer and edit the properties in the **Properties** window. This setting applies only to columns with a number-based data type, such as *int*.

Identity Increment
Shows the increment that will be added to the **Identity Seed** for each subsequent row. If you leave this cell blank, the value 1 will be assigned by default. To edit this property, type the new value directly.

Identity Seed
Shows the value assigned to the first row in the table. If you leave this cell blank, the value 1 will be assigned by default. To edit this property, type the new value directly.

Is Deterministic
Shows whether the data type of the selected column can be determined with certainty.

Is DTS-published
Shows whether the column is DTS-published.

Is Indexable
Shows whether the selected column can be indexed. For example, non-deterministic

computed columns cannot be indexed.

Is Merge-published
Shows whether the column is merge-published.

Is Not For Replication
Indicates whether original identity values are preserved during replication. To edit this property, click its value, expand the drop-down list, and choose another value.

Is Replicated
Shows whether this column is replicated in another location.

Is RowGuid
Indicates whether SQL Server uses the column as a ROWGUID. You can set this value to **Yes** only for a column with the data type of **uniqueidentifier**. To edit this property, click its value, expand the drop-down list, and choose another value.

Size
Shows the size in bytes allowed by column's data type. For example, a **nchar** data type may have a length of 10 (the number of characters) but it would have a size of 20 to account for Unicode character sets.

> **Note**
>
> The length of a **varchar(max)** data type varies for each row. sp_help returns (-1) as the length of **varchar(max)** column. Management Studio displays -1 as the column size.

Feedback

Was this page helpful? Yes No

Create Column Aliases (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can create aliases for column names to make it easier to work with column names, calculations, and summary values. For example, you can create a column alias to:

- Create a column name, such as "Total Amount," for an expression such as `(quantity * unit_price)` or for an aggregate function.
- Create a shortened form of a column name, such as `"d_id"` for `"discounts.stor_id."`

After you have defined a column alias, you can use the alias in a Select query to specify query output.

To create a column alias

1. In the **Criteria Pane**, locate the row containing the data column for which you want to create an alias, and if necessary, mark it for output. If the data column is not already in the grid, add it.
2. In the **Alias** column for that row, enter the alias. The alias must follow all naming conventions for SQL. If the alias name you enter contains spaces, the Query and View Designer automatically puts delimiters around it.

See Also

Add Columns to Queries (Visual Database Tools)
Sort and Group Query Results (Visual Database Tools)
Summarize Query Results (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Modify Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

Open a query to modify it in Query and View Designer.

To open a query

1. From the **View** menu, click **Solution Explorer**.
2. In Solution Explorer, right-click the query you want to open and click **Open**.
3. To modify the query in Query Designer, highlight the SQL statements, right-click the highlighted area and select **Design Query in Editor.**

Query Designer is now open and you can make modifications.

See Also

Run Queries (Visual Database Tools)
Work with Data in the Results Pane (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Draw Reflexive Relationships (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You create a reflexive relationship to link a column or columns in a table with another column or columns in the same table. For example, suppose the `employee` table has an `emp_id` column and a `mgr_id` column. Because each manager is also an employee, you relate these two columns by drawing a relationship line from the table to itself. This relationship ensures each manager ID that is added to the table matches an existing employee ID.

Before you create a relationship, you must first define a primary key or unique constraint for your table. You then relate the primary key column to a matching column. Once you create the relationship, the matching column becomes a foreign key of the table.

To draw a reflexive relationship

1. In your database diagram, click the row selector for the database column that you want to relate to another column and drag the pointer outside the table until a line appears.
2. Drag the line back to the selected table.
3. Release the mouse button. The **Tables and Columns** dialog box appears.
4. Select the foreign key column and the primary key table and column with which you want form a relationship.
5. Choose **OK** twice to create the relationship.

When you run queries against a table, you can use a reflexive relationship to create a self-join. For information about querying tables with joins, see Query with Joins (Visual Database Tools).

See Also

Query with Joins (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Edit Rows in the Results Pane (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

In many cases, you can edit the data in the Results Pane. For more detailed information about changing data see Rules for Updating Results (Visual Database Tools) and Work with Data in the Results Pane (Visual Database Tools).

To edit data in the Results pane

1. Run the query or view that has the data you want to edit.
2. Navigate to the cells containing the data you want to change.
3. Type in the new data.

   > **Note**
   >
   > To enter a null value type NULL in capital letters. The Results pane will format the word with italics to indicate it is a null value and not a string.

4. Save your changes by leaving the row.

See Also

Work with Data in the Results Pane (Visual Database Tools)
Rules for Updating Results (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Rules for Updating Results (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

In many cases, you can update the result set displayed in the Results Pane. However, in some cases you cannot.

In general, in order to update results, the Query and View Designer must have sufficient information to uniquely identify the row in the table. An example is if the query includes a primary key in the output list. In addition, you must have sufficient permission to update the database.

If your query is based on a view, you might be able to update it. The same guidelines apply, except that they apply to the underlying tables in the view, not just to the view itself.

> **Note**
>
> The Query and View Designer cannot determine in advance whether you can update a result set based on a view. Therefore, it displays all views, even though you might not be able to update them.

The following table summarizes specific instances in which you might and might not be able to update query results in the Results pane. In many cases, the database you are using dictates whether you can update query results.

Expand table

Query	Can results be updated?
Query based on one table with primary key in the output list	Yes (except as listed below).
Query based on a table with no unique index and without a primary key	Depends on query and database. Some databases allow updates if sufficient information is available to uniquely identify records.
Query based on multiple tables which are not joined	No.

Query	Can results be updated?
Query based on data marked as read-only in the database	No.
Query based on a view that involves one table with no constraints	Yes (except as listed below).
Query based on tables joined with a one-to-one relationship	Yes (except as listed below).
Query based on tables joined with a one-to-many relationship	Usually.
Query based on three or more tables in which there is a many-to-many relationship	No.
Query based on a table for which update permission is not granted	Can delete but not update.
Query based on a table for which delete permission is not granted	Can update but not delete.
Aggregate query	No.
Query based on a subquery that contains totals or aggregate functions	No.
Query that includes the DISTINCT keyword to exclude duplicate rows	No.
Query whose FROM clause includes a user-defined function that returns a table and the user-defined function contains multiple select statements	No.
Query whose FROM clause includes an inline user-defined function	Yes.

In addition, you might not be able to update specific columns in the query results. The following list summarizes specific types of columns that you cannot update in the Results pane.

- Columns based on expressions
- Columns based on scalar user-defined functions
- Rows or columns deleted by another user

- Rows or columns locked by another user (locked rows can usually be updated as soon as they are unlocked)
- Timestamp or BLOB columns

See Also

Design Queries and Views How-to Topics (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Count Rows in a Table (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can count rows in a table to determine:

- The total number of rows in a table, for example, a count of all the books in a `titles` table.
- The number of rows in a table that meet a specific condition, for example, the number of books by one publisher in a `titles` table.
- The number of values in a particular column.

When you count values in a column, nulls are not included in the count. For example, you might count the number of books in a `titles` table that have values in the `advance` column. By default, the count includes all values, not just unique values.

The procedures for all three types of counts are similar.

To count all the rows in a table

1. Be sure the table you want to summarize is already present in the Diagram pane.
2. Right-click the background of the Diagram pane, then choose **Add Group By** from the shortcut menu. The Query and View Designer adds a **Group By** column to the grid in the Criteria pane.
3. Select * **(All Columns)** in the rectangle representing the table or table-valued object.

   The Query and View Designer automatically fills the term **Count** into the **Group By** column in the Criteria pane and assigns a column alias to the column you are summarizing. You can replace this automatically generated alias with a more meaningful one. For more details, see Create Column Aliases (Visual Database Tools).
4. Run the query.

To count all the rows that meet a condition

1. Be sure the table you want to summarize is already present in the Diagram pane.
2. Right-click the background of the Diagram pane, then choose **Add Group By** from the shortcut menu. The Query and View Designer adds a **Group By** column to the grid in the Criteria pane.
3. Select ***(All Columns)** in the rectangle representing the table or table-structured object.

   The Query and View Designer automatically fills the term **Count** into the **Group By** column in the Criteria pane and assigns a column alias to the column you are summarizing. To create a more useful column heading in query output, see Create Column Aliases (Visual Database Tools).
4. Add the data column that you want to search, and then clear the check box in the **Output** column.

   The Query and View Designer automatically fills the term **Group By** into the **Group By** column of the grid.
5. Change **Group By** in the **Group By** column to **Where**.
6. In the **Filter** column for the data column to search, enter the search condition.
7. Run the query.

To count the values in a column

1. Be sure the table you want to summarize is already present in the Diagram pane.
2. Right-click the background of the Diagram pane, then choose **Add Group By** from the shortcut menu. The Query and View Designer adds a **Group By** column to the grid in the Criteria pane.
3. Add the column that you want to count to the Criteria pane.

   The Query and View Designer automatically fills the term **Group By** into the **Group By** column of the grid.
4. Change **Group By** in the **Group By** column to **Count**.

> **Note**
>
> To count only unique values, choose **Count Distinct**.

5. Run the query.

See Also

Sort and Group Query Results (Visual Database Tools)
Summarize Query Results (Visual Database Tools)

Feedback

Was this page helpful? Yes No

SQL Pane (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can use the SQL pane to create your own SQL statement, or you can use the Criteria pane and Diagram pane to create the statement, in which case the SQL statements will be created in the SQL pane. As you build your query, the SQL pane automatically updates and reformats for easy readability.

To open the SQL pane, first open Query and View Designer (with a database object selected in Server Explorer, from the **Database** menu, click **New Query**). Then from the **Query Designer** menu point to **Pane** and click **SQL**.

In the SQL pane you can:

- Create new queries by entering SQL statements.
- Modify the SQL statement created by the Query and View Designer based on settings you make in the Diagram and Criteria panes.
- Enter statements that take advantage of features specific to the database you are using.

> **Note**
>
> Be sure you know the rules for identifying database objects in the database you are using. For details, see the documentation for your database management system.

Statements in the SQL Pane

You can edit the current query directly in the SQL pane. When you move to another pane, the Query and View Designer automatically formats your statement, and then changes the Diagram and Criteria panes to match your statement.

If your statement cannot be represented in the Diagram and Criteria panes, and if those panes are visible, Query and View Designer displays an error and then offers you two choices:

- Ignore that the statement can not be represented in the Diagram and Criteria panes.

- Undo the change that can not be represented and revert to the most recent version of the SQL statement.

If you choose to ignore that the statement can not be represented in the Diagram and Criteria panes, the Query and View Designer dims the other panes to indicate that they no longer reflect the contents of the SQL pane.

You can continue to edit the statement and execute it as you would any SQL statement.

> **Note**
>
> If you enter a SQL statement, but then make further changes to the query by changing the Diagram and Criteria panes, the Query and View Designer rebuilds and redisplays the SQL statement. In some cases, this action results in a SQL statement that is constructed differently from the one you originally entered (though it will always yield the same results). This difference is particularly likely when you are working with search conditions that involve several clauses linked with AND and OR.

See Also

Create Queries (Visual Database Tools)
Run Queries (Visual Database Tools)
Design Queries and Views How-to Topics (Visual Database Tools)
Diagram Pane (Visual Database Tools)
Criteria Pane (Visual Database Tools)
Results Pane (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Sort in Ascending or Descending Order (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can sort query results in ascending or descending order on one or more of the columns in the result set by using the **ASC** or **DESC** keywords with the **ORDER BY** clause.

> **Note**
>
> The sort order is determined in part by the column's collation sequence. You can change the collation sequence in the Collation dialog box.

The following procedure assumes that you have a query open in Query and View Designer that uses the ORDER BY clause to sort one or more columns.

To specify or change the order in which results are sorted

1. In the Criteria pane, click the **Sort Type** field for the column that you want to reorder.
2. Choose **Ascending** or **Descending** to specify the sort order for the column.

Notice that as you work in the Criteria pane, your query's UNION clause changes to match your most recent actions.

> **Note**
>
> When sorting results by more than one column, specify the order in which columns are searched relative to each other by using the Sort Order column. For more information, see Sort Multiple Columns in Queries (Visual Database Tools).

See Also

Sort and Group Query Results (Visual Database Tools)
Summarize Query Results (Visual Database Tools)
Design Queries and Views How-to Topics (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Conventions for Combining Search Conditions in the Criteria Pane (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can create queries that include any number of search conditions, linked with any number of AND and OR operators. A query with a combination of AND and OR clauses can become complex, so it is helpful to understand how such a query is interpreted when you execute it, and how such a query is represented in the Criteria Pane and SQL Pane.

> **Note**
>
> For details about search conditions that contain only one AND or OR operator, see Specify Multiple Search Conditions for One Column (Visual Database Tools) and Specify Multiple Search Conditions for Multiple Columns (Visual Database Tools).

Below you will find information about:

- The precedence of AND and OR in queries that contain both.
- How the conditions in AND and OR clauses relate logically to one another.
- How the Query and View Designer represents in the Criteria Pane queries that contain both AND and OR.

To help you understand the discussion below, imagine that you are working with an `employee` table containing the columns `hire_date`, `job_lvl`, and `status`. The examples assume that you need to know information such as how long an employee has worked with the company (that is, what the employee's hire date is), what type of job the employee performs (what the job level is), and the employee's status (for example, retired).

Precedence of AND and OR

When a query is executed, it evaluates first the clauses linked with AND, and then those linked with OR.

> **Note**
>
> The NOT operator takes precedence over both AND and OR.

For example, to find either employees who have been with the company for more than five years in lower-level jobs or employees with middle-level jobs without regard for their hire date, you can construct a WHERE clause such as the following:

```
WHERE
   hire_date < '01/01/95' AND
   job_lvl = 100 OR
   job_lvl = 200
```

To override the default precedence of AND over OR, you can put parentheses around specific conditions in the SQL pane. Conditions in parentheses are always evaluated first. For example, to find all employees who have been with the company more than five years in either lower or middle-level jobs, you can construct a WHERE clause such as the following:

```
WHERE
   hire_date < '01/01/95' AND
   (job_lvl = 100 OR job_lvl = 200)
```

> **Tip**
>
> It is recommended that, for clarity, you always include parentheses when combining AND and OR clauses instead of relying on the default precedence.

How AND Works with Multiple OR Clauses

Understanding how AND and OR clauses are related when combined can help you construct and understand complex queries in the Query and View Designer.

If you link multiple conditions using AND, the first set of conditions linked with AND applies to all the conditions in the second set. In other words, a condition linked with AND to another condition is distributed to all the conditions in the second set. For

example, the following schematic representation shows an AND condition linked to a set of OR conditions:

```
A AND (B OR C)
```

The representation above is logically equivalent to the following schematic representation, which shows how the AND condition is distributed to the second set of conditions:

```
(A AND B) OR (A AND C)
```

This distributive principle affects how you use the Query and View Designer. For example, imagine that you are looking for all employees who have been with the company more than five years in either lower or middle-level jobs. You enter the following WHERE clause into the statement in the SQL pane:

```
WHERE (hire_date < '01/01/95' ) AND
   (job_lvl = 100 OR job_lvl = 200)
```

The clause linked with AND applies to both clauses linked with OR. An explicit way to express this is to repeat the AND condition once for each condition in the OR clause. The following statement is more explicit (and longer) than the previous statement, but is logically equivalent to it:

```
WHERE    (hire_date < '01/01/95' ) AND
  (job_lvl = 100) OR
  (hire_date < '01/01/95' ) AND
  (job_lvl = 200)
```

The principle of distributing AND clauses to linked OR clauses applies no matter how many individual conditions are involved. For example, imagine that you want to find higher or middle-level employees who have been with the company more than five years or are retired. The WHERE clause might look like this:

```
```

```
WHERE
   (job_lvl = 200 OR job_lvl = 300) AND
   (hire_date < '01/01/95' ) OR (status = 'R')
```

After the conditions linked with AND have been distributed, the WHERE clause will look like this:

```
WHERE
   (job_lvl = 200 AND hire_date < '01/01/95' ) OR
   (job_lvl = 200 AND status = 'R') OR
   (job_lvl = 300 AND hire_date < '01/01/95' ) OR
   (job_lvl = 300 AND status = 'R')
```

How Multiple AND and OR Clauses Are Represented in the Criteria Pane

The Query and View Designer represents your search conditions in the Criteria Pane. However, in some cases that involve multiple clauses linked with AND and OR, the representation in the Criteria Pane might not be what you expect. In addition, if you modify your query in the Criteria Pane or Diagram Pane, you might find that your SQL statement has been changed from what you entered.

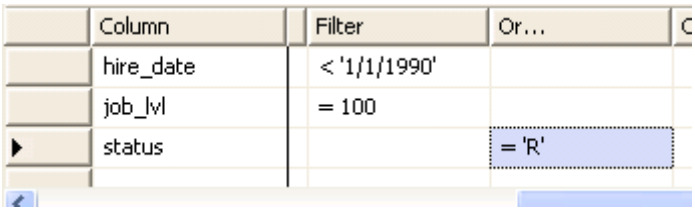
In general, these rules dictate how AND and OR clauses appear in the Criteria Pane:

- All conditions linked with AND appear in the **Filter** grid column or in the same **Or...** column.
- All conditions linked with OR appear in separate **Or...** columns.
- If the logical result of a combination of AND and OR clauses is that the AND is distributed into several OR clauses, the Criteria Pane represents this explicitly by repeating the AND clause as many times as necessary.

For example, in the SQL pane you might create a search condition such as the following, in which two clauses linked with AND take precedence over a third one linked with OR:

```
WHERE (hire_date < '01/01/95' ) AND
  (job_lvl = 100) OR
  (status = 'R')
```

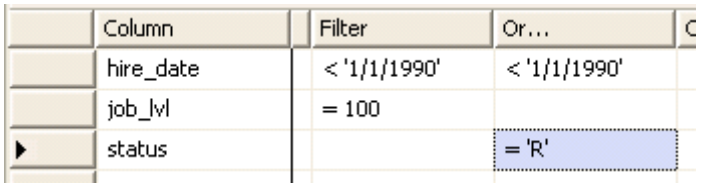
The Query and View Designer represents this WHERE clause in the Criteria Pane as follows:



However, if the linked OR clauses take precedence over an AND clause, the AND clause is repeated for each OR clause. This causes the AND clause to be distributed to each OR clause. For example, in the SQL pane you might create a WHERE clause such as the following:

```
WHERE (hire_date < '01/01/95' ) AND
  ( (job_lvl = 100) OR
  (status = 'R') )
```

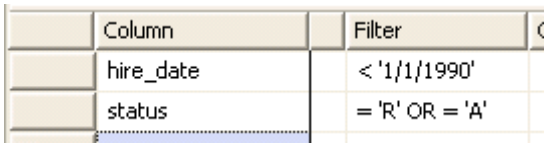
The Query and View Designer represents this WHERE clause in the Criteria Pane as follows:



If the linked OR clauses involve only one data column, the Query and View Designer can place the entire OR clause into a single cell of the grid, avoiding the need to repeat the AND clause. For example, in the SQL pane you might create a WHERE clause such as the following:

```
WHERE (hire_date < '01/01/95' ) AND
  ((status = 'R') OR (status = 'A'))
```

The Query and View Designer represents this WHERE clause in the Criteria Pane as follows:



If you make a change to the query (such as changing one of the values in the Criteria Pane), the Query and View Designer recreates the SQL statement in the SQL pane. The recreated SQL statement will resemble the Criteria Pane display rather than your original statement. For example, if the Criteria Pane contains distributed AND clauses, the resulting statement in the SQL pane will be recreated with explicit distributed AND clauses. For details, see "How AND Works with Multiple OR Clauses" earlier in this topic.

See Also

Specify Search Criteria (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Modify Join Operators (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

By default, the Query and View Designer joins tables using an equal sign (an equijoin), which matches values in the two join columns. If you want, you can change the operator used to compare values in the join columns.

To modify join operators

1. In the Diagram pane, right-click the join line you want to modify, and then choose **Properties** from the shortcut menu.
2. In the Property window, click **Join Condition and Type** and click the **ellipses (...)** to the right of the property.
3. In the **Join** dialog box, select a new operator.

See Also

Join Tables Automatically (Visual Database Tools)
Join Tables Manually (Visual Database Tools)
Query with Joins (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Remove Joins (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

If you do not want tables to be joined via an inner join or an outer join, you can remove the join between them. For example, you might remove a join that the Query and View Designer has been created automatically between two tables.

> **Note**
>
> Removing a join from a query does not alter the underlying relationship in the database.

If two joined tables are part of your query and you remove all join conditions between them, the resulting query becomes the product of both tables - that is, it becomes a CROSS JOIN.

To remove a join

- In the Diagram pane, select the join line for the join to remove, and then press the DELETE key. You can select and delete multiple join lines at one time.

The Query and View Designer removes the join line and alters the statement in the SQL pane.

See Also

Join Tables Automatically (Visual Database Tools)
Join Tables Manually (Visual Database Tools)
Query with Joins (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Table Properties (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

These properties appear in the Properties window when you right-click in Table Designer and select Properties. Unless otherwise noted, you can edit these properties in the Properties window when the table is selected.

> **Note**
>
> If the table is published for replication, you must make schema changes using the Transact-SQL statement ALTER TABLE or SQL Server Management Objects (SMO). When schema changes are made using the Table Designer or the Database Diagram Designer, it attempts to drop and recreate the table. You cannot drop published objects, therefore the schema change will fail.

Show Table Properties in Table Designer

> **Note**
>
> The properties in this topic are ordered by category rather than alphabet.

Identity Category
Expands to show properties for **Name**, **Description**, and **Schema**.

Name
Displays the name of the table. To edit the name, type in the text box.

> **Caution**
>
> If existing queries, views, user-defined functions, stored procedures, or programs refer to the table, the name modification will make these objects invalid.

Database Name
Shows the name of the data source of the selected table.

Description
Shows a description of the selected table. To see the entire description, or to edit it, click

the description and then click the ellipses **(...)** to the right of the property.

Schema
Shows the name of the schema to which this table belongs. (Applies only to Microsoft SQL Server.)

Server Name
Shows the name of the server for the data source.

Table Designer Category
Expands to show properties for **Identity Column**, **Is Indexable**, and **Is Replicated**.

Identity Column
Shows the column used as the table's identity column. To change the identity column, choose from the drop-down list. Only columns of a numeric data type will show in the list.

Is Indexable
Shows whether the table can be indexed. If the table is not indexable it may be because you are not the table owner or because the table contains columns with data types of text, ntext, or image.

Is Replicated
Shows whether the table is replicated in another location.

Regular Data Space Specification Category
Expands to show properties for **(Data Space Type)**, **Filegroup or Partition Scheme Name**, and **Partition Column List**.

(Data Space Type)
Shows whether this table is stored using a filegroup or partition scheme.

Filegroup or Partition Scheme Name
Shows the name of the filegroup or partition scheme.

Partition Column List
Provides access to the **Partition Column List** dialog box.

Row GUID Column
Shows the column used by Microsoft SQL Server as the table's ROWGUID column. To change the ROWGUID column, choose from the drop-down list. (Applies only to SQL Server 7.0 or higher.)

Text/Image Filegroup
Provides a drop-down list from which you can choose the filegroup for columns that

have text or image data types. If the table is stored using a partition scheme, leave this field blank.

See Also

Design Tables

Feedback

Was this page helpful? Yes No

Development, Test, and Production Databases (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

If you have two databases with identical structure, you can make changes in one database and propagate those changes to the other. For example, if you have a personal development database and a group-wide test database, you can modify the development database, then propagate those changes to the test database.

To accomplish this, perform all the modifications in a single session with the development database, create a Change Script of your session and later run the script on the test database.

See Also

Multiuser Environments (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Summarize Query Results (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

When you create aggregate queries, certain logical principles apply. For example, you cannot display the contents of individual rows in a summary query. The Query and View Designer helps you comply with these principles in the way the Diagram pane and Criteria pane behave.

By understanding the principles of aggregate queries and the Query and View Designer's behavior, you can create logically correct aggregate queries. The overriding principle is that aggregate queries can result only in summary information. Thus, most of the principles that follow describe the ways that you can reference individual data columns within an aggregate query.

In This Section

Work with Columns in Aggregate Queries (Visual Database Tools)
Describes concepts about grouping and summarizing columns with the GROUP BY, WHERE, and HAVING clauses.

Count Rows in a Table (Visual Database Tools)
Provides steps for counting the number of rows in a table or the number of rows in a table that meet a set of criteria.

Summarize or Aggregate Values for All Rows in a Table (Visual Database Tools)
Provides steps for summarizing all rows rather than for a set of grouped rows.

Summarize or Aggregate Values Using Custom Expressions (Visual Database Tools)
Provides steps for using expressions to summarize or aggregate rather than using the predefined clauses.

Related Sections

Design Queries and Views How-to Topics (Visual Database Tools)
Provides links to topics covering how to use the Query and View Designer.

Feedback

Was this page helpful? Yes No

Upgrade Database Diagrams from Previous Editions (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

Database diagrams created on previous editions of SQL Server databases are upgraded automatically when database diagramming is set up on SQL Server. For more steps for setting up diagramming see Set Up Database Diagram Designer.

To upgrade legacy database diagrams

1. From Object Explorer, expand the database.
2. Expand the Database Diagram node under the database.
3. Select **Yes** when prompted if you want to set up database diagramming.

   This will upgrade diagrams stored on the database.

See Also

Understand Database Diagram Ownership
Set Up Database Diagram Designer

Feedback

Was this page helpful? Yes No

Create and update database tables

Article • 02/13/2025

Applies to: SQL Server

The Table Designer is a visual tool where you design and visualize database tables. To create, edit, or delete tables, columns, keys, indexes, relationships, and constraints, use the Table Designer in Download SQL Server Management Studio (SSMS) or Azure Data Studio (ADS).

Create a table

1. Right-click the **Tables** node in your database and select **New** > **Table** in SSMS, or **New Table** in ADS.
2. Add columns to your table, specifying name, data type, and whether `NULL` values are allowed for each column.
3. Other table properties can be configured in Azure Data Studio including keys, constraints, and indexes. For more information, see Overview of the table designer in Azure Data Studio.
4. In SSMS, close the designer and specify a table name to save your changes. In Azure Data Studio, ensure the table name is updated in the **Table name** field, then select the **Publish Changes** icon, or close the designer and save the changes to create the table.

Update a table

1. Right-click the table under the **Tables** node of your database and select **Design**.
2. Update the desired table settings.
3. Close the designer and save your changes, or use the **Publish Changes** icon in Azure Data Studio.

Permissions

In order to create a table, you must have `CREATE TABLE` permissions in the database and `ALTER` permission on the schema in which the table is being created. For more information, see CREATE TABLE (Transact-SQL).

To alter a table, you must have `ALTER` permissions on the table. For more information, see ALTER TABLE (Transact-SQL).

Related content

- Tables
- Table Properties (Visual Database Tools)
- Column Properties (Visual Database Tools)
- Add Columns to a Table (Database Engine)
- Primary and Foreign Key Constraints
- Indexes
- Data types (Transact-SQL)
- Create a database and add tables in Visual Studio

Feedback

Was this page helpful? Yes No

Permissions and Visual Database Tools (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

Permissions, which allow users access to data source objects, are granted in the data source, not in Visual Database Tools. The access permissions granted on data source objects is controlled by your installation's security policy. If you do not have permission to access certain database objects, you should contact your database administrator.

If the database permissions let you view but not modify the database structure, you can use change scripts to design your modifications without transmitting them to the database.

See Also

Multiuser Environments (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Add Related Tables to Diagrams (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

You can quickly add to a database diagram all of the tables that are involved in a relationship with a selected table.

To add related tables to a diagram

1. Add the table with relationships to a database diagram.
2. Right-click the table and from the shortcut menu and click **Add Related Tables**.

Feedback

Was this page helpful? Yes No

Visual Database Tool Designers

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

The Visual Database Tools are a combination of design tools you can use to work with a data source. You can use them to create queries, design or modify a database structure, or update data. The tools are Database Diagram Designer, Table Designer, and Query and View Designer.

Properties Window

The properties window is not specific to Visual Database Tools, but it is where many modifications can be made. It shows the properties for the item currently selected (like a table) and allows you to edit those properties (everything from the properties name to the collation of a column). Some properties can be seen in the properties window but must be modified in a different tool.

Database Diagram Designer

Database Diagram Designer provides a window where you can visually create, edit, and show the tables and relationships in a database.

To display Database Diagram Designer, open an existing diagram or right-click the Database node in Object Explorer and choose **New Database Diagram** from the drop-down menu.

Once the designer is open, the **Database Diagram** menu appears in the main menu. This menu is the access point to the designer's special features.

> **Note**
>
> This designer works with Microsoft SQL Server databases.
>
> This version of Visual Database Tools does not support Microsoft SQL Server version 7 and earlier.

Table Designer

Table Designer is a visual tool allowing you to design and visualize a single table in a Microsoft SQL Server database to which you are connected.

Table Designer has two panes. The upper area shows a grid; each row of the grid describes one database column. For each database column, the grid displays its fundamental attributes: column name, data type, and nulls-allowed setting.

The lower area of Table Designer, the Column Properties tab, shows additional attributes for whichever column is highlighted in the upper area.

From Table Designer, you can also right-click in the grid section to access dialog boxes through which you can create and modify relationships, constraints, indexes, and keys for the table.

To display Table Designer, open an existing table, or right-click the **Tables** node in Object Explorer, and choose **Add New Table** from the drop-down menu.

Once the designer is open, the Table Designer menu appears in the main menu. This menu is the access point to the designer's special features.

> ⓘ **Note**
>
> This designer works with Microsoft SQL Server databases.
>
> This version of Visual Database Tools does not support Microsoft SQL Server version 7 and earlier.

Query and View Designer

Query and View designer is actually two tools that work in very similar ways. Some of the major differences are:

- Views are saved with the database while a query is saved with a Visual Studio database project.
- Query Designer works with nearly any data source, where View Designer works only with SQL Server.
- Query Designer allows you to design SELECT, INSERT, UPDATE and DELETE DML statements, while views can only contain SELECT statements.

View Designer

View Designer allows you to design and visualize an existing view or create a new one in a Microsoft SQL Server database to which you are connected.

View Designer has four panes: the Diagram pane, the Criteria pane, the SQL pane, and the Results pane. For more detailed information on each of these panes, see Query and View Designer Tools (Visual Database Tools).

To display View Designer, open an existing view or right-click the **View** node in Object Explorer and choose **Add New View** from the drop-down menu.

Once the designer is open, the **Query Designer** menu appears in the main menu. This menu is the access point to the designer's special features.

> **Note**
>
> This designer works with Microsoft SQL Server databases.
>
> This version of Visual Database Tools does not support Microsoft SQL Server version 7 and earlier.

See Also

Design Database Diagrams (Visual Database Tools)
Design Tables (Visual Database Tools)
Design Queries and Views How-to Topics (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Specify the TOP Clause in Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

The TOP clause returns only the first *n* or *n percent* rows from a query. The TOP clause is useful when you want to inspect a portion of your results to find out if your query does what you expect it to do, without taking the resources necessary to return all of the query results.

To specify the TOP clause in queries

1. Open a query from Solution Explorer or create a new one.
2. From the **View** menu, click **Properties Window**.
3. In the **Properties Window**, locate and expand the **Top Specification** property.
4. Click the **(Top)** child property and set it to **Yes**.
5. In the **Expression** child property, type an expression that has a numeric result (for example, "10" or "2*5").
6. Click the **Percent** child property and indicate whether or not to treat the **Expression** property as a percentage of all rows returned (Yes) or as the absolute number of rows returned (No).
7. If your query uses the ORDER BY clause, click the **With Ties** child property and choose **Yes** to display all rows in a group if only part of the group is included or **No** to truncate them.

As you work through the preceding procedure, notice that the TOP clause, displayed in the SQL pane, changes to reflect the current settings of the properties.

> **Note**
>
> You can also change the values of the child properties of the **Top Specification** by editing the TOP clause in the SQL pane.

See Also

Design Queries and Views How-to Topics (Visual Database Tools)
Query Properties (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Include or Exclude Rows (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

To restrict the number of rows a SELECT query should return, you create search conditions or filter criteria. In SQL, search conditions appear in the WHERE clause of the statement, or if you are creating an aggregate query, in the HAVING clause.

> **Note**
>
> You can also use search conditions to indicate which rows are affected by an Update, Insert Results, Insert Values, Delete, or Make Table query.

When the query runs, the Database Engine examines and applies the search condition to each row in the tables you are searching. If the row meets the condition, it is included in the query. For example, a search condition that would find all the employees in a particular region might be:

```
region = 'UK'
```

To establish the criteria for including a row in a result, you can use multiple search conditions. For example, the following search criterion consists of two search conditions. The query includes a row in the result set only if that row satisfies both of the conditions.

```
region = 'UK' AND product_line = 'Housewares'
```

You can combine these conditions with AND or OR. The previous example uses AND. In contrast, the following criterion uses OR. The result set will include any row that satisfies either or both of the search conditions:

```
region = 'UK' OR product_line = 'Housewares'
```

You can even combine search conditions on a single column. For example, the following criterion combines two conditions on the region column:

```
region = 'UK' OR region = 'US'
```

For details about combining search conditions, see the following topics:

- Conventions for Combining Search Conditions in the Criteria Pane (Visual Database Tools)
- Specify Multiple Search Conditions for One Column (Visual Database Tools)
- Specify Multiple Search Conditions for Multiple Columns (Visual Database Tools)
- Combine Conditions When AND Has Precedence (Visual Database Tools)
- Combine Conditions When OR Has Precedence (Visual Database Tools)

Examples

Here are some examples of queries using various operators and row criteria:

- **Literal** A single text, numeric, date, or logical value. The following example uses a literal to find all rows for employees in the United Kingdom:

  ```
  WHERE region = 'UK'
  ```

- **Column reference** Compares the values in one column with the values in another. The following example searches a `products` table for all rows in which the value of the production cost is lower than the shipping cost:

  ```
  WHERE prod_cost < ship_cost
  ```

- **Function** A reference to a function that the database back-end can resolve to calculate a value for the search. The function can be a function defined by the database server or a user-defined function that returns a scalar value. The following example searches for orders placed today (the GETDATE( ) function returns the current date):

```
WHERE order_date = GETDATE()
```

- **NULL** The following example searches an `authors` table for all authors who have a first name on file:

```
WHERE au_fname IS NOT NULL
```

- **Calculation** The result of a calculation that can involve literals, column references, or other expressions. The following example searches a `products` table to find all rows in which the retail sales price is more than twice the production cost:

```
WHERE sales_price > (prod_cost * 2)
```

See Also

Design Queries and Views How-to Topics (Visual Database Tools)
Specify Search Criteria (Visual Database Tools)
Query with Parameters (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Stop a Query (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

If you see that the query is taking too long or is not returning the results you expect, you can stop the query if it has not already finished.

To stop a query

- Right-click anywhere in the Results pane, on the shortcut menu point to **Pane**, and then click **Clear Results**.

See Also

Clear Query Results (Visual Database Tools)
Run Queries (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Show Column Names in the Diagram Pane (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can choose to hide or show the column names of tables in the **Diagram Pane** of the Query and View Designer.

Show Column Names

1. Right-click the table in the **Diagram Pane** of the Query and View Designer.
2. Choose **Column Names** from the shortcut menu.

Hide Column Names

1. Right-click the table in the **Diagram Pane** of the Query and View Designer.
2. Choose **Name Only** from the shortcut menu.

See Also

Design Queries and Views How-to Topics (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Create UNION Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

The UNION keyword enables you to include the results of two SELECT statements in one resulting table. All rows returned from either SELECT statement are combined into the result of the UNION expression. For examples, see SELECT Examples (Transact-SQL).

> **Note**
>
> The Diagram pane can only display one SELECT clause. Therefore, when you are working with a UNION query, Query Designer hides the Table Operations pane.

To create a Merged SELECT query

1. Open a query or create a new one.
2. In the SQL pane, type a valid UNION expression.

   The following example is a valid UNION expression.

   ```
   SELECT ProductModelID, Name
   FROM Production.ProductModel
   UNION
   SELECT ProductModelID, Name
   FROM dbo.Gloves;
   ```

3. On the **Query Designer** menu, click **Execute SQL** to run the query.

   Your UNION query is now formatted by Query Designer.

See Also

Supported Query Types
Design Queries and Views How-to Topics
Perform Basic Operations with Queries
UNION (Transact-SQL)

Feedback

Was this page helpful? Yes No

Open the Query and View Designer (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance

The Query and View Designer opens when you open the definition of a view, show the results for a query or view, or create or open a query. It consists of four separate panes:

- The Diagram pane presents a graphic display of the tables or table-valued objects you have selected from the data connection. It also shows any join relationships among them.
- The Criteria pane allows you to specify query options - such as which data columns to display, how to order the results, and what rows to select - by entering your choices into a spreadsheet-like grid.
- You can use the SQL pane to create your own SQL statement, or you can use the Criteria pane and Diagram pane to create the statement, in which case the SQL statements will be created in the SQL pane. As you build your query, the SQL pane automatically updates and reformats to be easily read.
- The Results pane shows the results of the most recently executed Select query. (The results of other query types are displayed in message boxes.)
- These panes are useful for working with both queries and views.
- When you open a view or query some or all of the panes open with it. Which ones open depend on the settings in the **Options** dialog box and what database management system you're connected to. The default is that all four open.

To open the Query and View Designer for a view

1. In Object Explorer, right-click the view you want to open and click **Design** or **Open View**.

   If you chose **Design**, the Query and View Designer panes open as dictated by the options selected in the **Options** dialog box. If you chose **Open View**, only the Results pane opens by default.

To open the Query and View Designer for an existing query

1. In Solution Explorer, expand the **Queries** folder.
2. Double-click the query you want to open.
3. Highlight the query statement(s), right-click the highlighted area and click **Design Query in Editor**.

See Also

Design Queries and Views How-to Topics (Visual Database Tools)
Query and View Designer Tools (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Combine Conditions When OR Has Precedence (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

To link conditions with OR and give them precedence over conditions linked with AND, you must repeat the AND condition for each OR condition.

For example, imagine that you want to find all employees who have been with the company more than five years and have lower-level jobs or are retired. This query requires three conditions, a single condition linked to two additional conditions with AND:

- Employees with a hire date earlier than five years ago, and
- Employees with a job level of 100 or whose status is "R" (for retired).

The following procedure illustrates how to create this type of query in the Criteria pane.

To combine conditions when OR has precedence

1. In the Criteria pane, add the data columns you want to search. If you want to search the same column using two or more conditions linked with AND, you must add the data column name to the grid once for each value you want to search.
2. Create the conditions to be linked with OR by entering the first one into the **Filter** grid column and the second (and subsequent ones) into separate **Or...** columns. For example, to link conditions with OR that search the `job_lvl` and `status` columns, enter `= 100` in the **Filter** column for `job_lvl` and `= 'R'` in the **Or...** column for `status`.

   Entering these values in the grid produces the following WHERE clause in the statement in the SQL pane:

   ```
   WHERE (job_lvl = 100) OR (status = 'R')
   ```

3. Create the AND condition by entering it once for each OR condition. Place each entry in the same grid column as the OR condition it corresponds to. For example,

to add an AND condition that searches the `hire_date` column and applies to both OR conditions, enter `< '1/1/91'` in both the Criteria column and the **Or...** column.

Entering these values in the grid produces the following WHERE clause in the statement in the SQL pane:

```
WHERE (job_lvl = 100) AND
  (hire_date < '01/01/91' ) OR
  (status = 'R') AND
  (hire_date < '01/01/91' )
```

> **Tip**
>
> You can repeat an AND condition by adding it once, and then using the **Cut** and **Paste** commands from the **Edit** menu to repeat it for other OR conditions.

The WHERE clause created by the Query and View Designer is equivalent to the following WHERE clause, which uses parentheses to specify the precedence of OR over AND:

```
WHERE (job_lvl = 100 OR status = 'R') AND
   (hire_date < '01/01/91')
```

> **Note**
>
> If you enter the search conditions in the format shown immediately above in the SQL Pane, but then make a change to the query in the Diagram or Criteria panes, the Query and View Designer recreates the SQL statement to match the form with the AND condition explicitly distributed to both OR conditions.

See Also

Conventions for Combining Search Conditions in the Criteria Pane (Visual Database Tools)
Specify Search Criteria (Visual Database Tools)

Feedback

Was this page helpful? Yes No

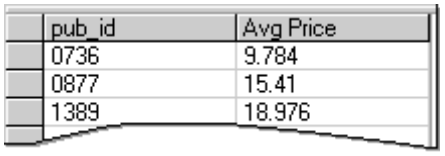
Group Rows in Query Results (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

If you want to create subtotals or show other summary information for subsets of a table, you create groups using an aggregate query. Each group summarizes the data for all the rows in the table that have the same value.

For example, you might want to see the average price of a book in the `titles` table, but break the results down by publisher. To do so, you group the query by publisher (for example, `pub_id`). The resulting query output might look like this:



pub_id	Avg Price
0736	9.784
0877	15.41
1389	18.976

When you group data, you can display only summary or grouped data, such as:

- The values of the grouped columns (those that appear in the GROUP BY clause). In the example above, `pub_id` is the grouped column.
- Values produced by aggregate functions such as SUM( ) and AVG( ). In the example above, the second column is produced by using the AVG( ) function with the `price` column.

You cannot display values from individual rows. For example, if you group only by publisher, you cannot also display individual titles in the query. Therefore, if you add columns to the query output, the Query and View Designer automatically adds them to the GROUP BY clause of the statement in the SQL pane. If you want a column to be aggregated instead, you can specify an aggregate function for that column.

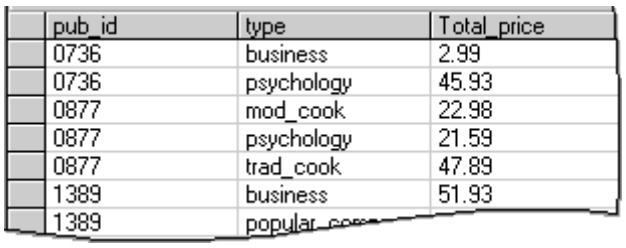
If you group by more than one column, each group in the query shows the aggregate values for all grouping columns.

For example, the following query against the `titles` table groups by publisher (`pub_id`) and also by book type (`type`). The query results are ordered by publisher and show summary information for each different type of book that the publisher produces:

```
```

```
SELECT pub_id, type, SUM(price) Total_price
FROM titles
GROUP BY pub_id, type
```

The resulting output might look like this:



pub_id	type	Total_price
0736	business	2.99
0736	psychology	45.93
0877	mod_cook	22.98
0877	psychology	21.59
0877	trad_cook	47.89
1389	business	51.93
1389	popular_comp	

To group rows

1. Start the query by adding the tables you want to summarize to the Diagram pane.

2. Right-click the background of the Diagram pane, then choose **Add Group By** from the shortcut menu. The Query and View Designer adds a **Group By** column to the grid in the Criteria pane.

3. Add the column or columns you want to group to the Criteria pane. If you want the column to appear in the query output, be sure that the **Output** column is selected for output.

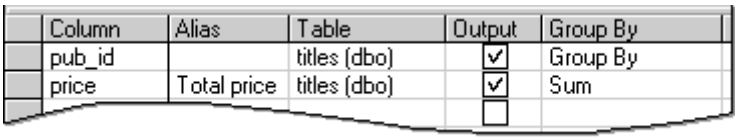
   The Query and View Designer adds a GROUP BY clause to the statement in the SQL pane. For example, the SQL statement might look like this:

   ```
   SELECT pub_id
   FROM titles
   GROUP BY pub_id
   ```

4. Add the column or columns you want to aggregate to the Criteria pane. Be sure that the column is marked for output.

5. In the **Group By** grid cell for the column that is going to be aggregated, select the appropriate aggregate function.

   The Query and View Designer automatically assigns a column alias to the column you are summarizing. You can replace this automatically generated alias with a more meaningful one. For more details, see Create Column Aliases.



Column	Alias	Table	Output	Group By
pub_id		titles (dbo)	☑	Group By
price	Total price	titles (dbo)	☑	Sum
			☐	

The corresponding statement in the **SQL** pane might look like this:

```
SELECT   pub_id, SUM(price) AS Totalprice
FROM     titles
GROUP BY pub_id
```

See Also

Sort and Group Query Results

Feedback

Was this page helpful? Yes No

Sort and Group Query Results (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can create a query result in which each result row corresponds to an entire group of rows from the original data.

To learn the details about creating such queries, see the topics listed in the following table.

In This Section

Sort Rows (Visual Database Tools)
Describes the various ways in which you can sort rows and why you would use them.

Collapse Groups of Rows (Visual Database Tools)
Describes the various ways to collapse rows, such as calculating or eliminating duplicates.

Sort with ORDER BY (Visual Database Tools)
Provides steps for returning results in a specified order.

Sort in Ascending or Descending Order (Visual Database Tools)
Provides steps for changing or setting sort direction.

Sort Multiple Columns in Queries (Visual Database Tools)
Provides steps for setting the order of results sets for multiple columns.

Group Rows in Query Results (Visual Database Tools)
Provides steps for creating subsets of summary information by organizing data into groups.

Specify Conditions for Groups (Visual Database Tools)
Provides steps for creating search conditions that apply to groups of rows.

Reorder Output Columns (Visual Database Tools)
Provides steps for changing current sorting settings.

Related Sections

Summarize Query Results (Visual Database Tools)

Provides links to topics on summarizing query results.

Perform Basic Operations with Queries (Visual Database Tools)

Provides links to topics covering the most common query tasks.

Design Queries and Views How-to Topics (Visual Database Tools)

Provides links to topics covering how to use the Query and View Designer.

Feedback

Was this page helpful? Yes No

Exclude Duplicate Rows (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

If you want to see only unique values in a result set, you can specify that you want to exclude duplicates from the result set.

To exclude duplicate rows from the result set

1. Right-click the background of the Diagram pane, then choose **Properties** from the shortcut menu.
2. In the Property window, click **Distinct values** and set the value to **Yes**.

   The Query and View Designer inserts the keyword DISTINCT in front of the list of display columns in the SQL statement.

   > **Note**
   >
   > If you use the DISTINCT keyword you may not be able to modify the result set in the results pane.

See Also

Specify Search Criteria (Visual Database Tools)
Sort and Group Query Results (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Remove Columns from Query Results (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

If you are using a column in a Select query but do not want to display it in the result set (that is, you do not want it in the query's select list), you can remove it from output. After you remove the column from the query's output, you can still use it in search conditions or as a sorting field.

> **Note**
>
> If you want to remove a column from the query altogether, see Remove Columns from Queries (Visual Database Tools).

To remove a column from the query output

- In the **Criteria Pane**, clear the check box in the **Output** column for the data column you want to remove. (If you want to add the column back to the query output, you can check the **Output** column again.)

  -or-

- Remove the column from the output list in the SQL pane.

See Also

Add Columns to Queries (Visual Database Tools)
Remove Columns from Queries (Visual Database Tools)
Sort and Group Query Results (Visual Database Tools)
Summarize Query Results (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Rename Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can rename any query that is not currently open for editing. The following procedure assumes that the query you want to rename is closed.

To rename a query

1. From the **View** menu, choose **Solution Explorer**.
2. In Solution Explorer, right-click the query you want to rename and click **Rename** in the shortcut menu that appears.
3. Type a new name for the query and then press Enter.

See Also

Design Queries and Views How-to Topics (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Delete Rows in the Results Pane (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

Delete rows in the Results pane if you want to delete records in the database. If you want to delete all of the rows you can use a Delete query. For more information see Create Delete Queries (Visual Database Tools). If you only want to remove rows from the Results pane, change the criteria for the query. For more information see Specify Search Criteria (Visual Database Tools).

To delete a row or rows

1. Select the box to the left of the row or rows you want to delete in the Results pane.
2. Press DELETE.
3. In the message box asking for confirmation, click **Yes**.

> **Caution**
>
> Rows you delete in this way are permanently removed from the database and cannot be recalled.

> **Note**
>
> If any of the selected rows can't be deleted from the database, none of them will be deleted and you will receive a message telling you which rows can't be deleted.

See Also

Create Delete Queries (Visual Database Tools)
Specify Search Criteria (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Join Tables Manually (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

When you add two (or more) tables to a query, the Query and View Designer attempts to join them based on common data or on information stored in the database about how tables are related. For details, see Join Tables Automatically (Visual Database Tools). However, if the Query and View Designer has not joined the tables automatically, or if you want to create additional join conditions between tables, you can join tables manually.

You can create joins based on comparisons between any two columns, not just columns that contain the same information. For example, if your database contains two tables, `titles` and `roysched`, you can compare values in the `ytd_sales` column of the `titles` table against the `lorange` and `hirange` columns in the `roysched` table. Creating this join would allow you to find titles for which the year-to-date sales falls between the low and high ranges for the royalty payments.

> **Tip**
>
> Joins work fastest if the columns in the join condition have been indexed. In some cases, joining on unindexed columns can result in a slow query.

To manually join tables or table-structured objects

1. Add to the Diagram pane the objects you want to join.
2. Drag the name of the join column in the first table or table-structured object and drop it onto the related column in the second table or table-structured object. You cannot base a join on **text**, **ntext**, or i**mage** columns.

   > **Note**
   >
   > The join columns must be of the same (or compatible) data types. For example, if the join column in the first table is a date, you must relate it to a date column in the second table. On the other hand, if the first join column is an integer, the related join column must also be of an integer data type, but it

can be a different size. The Query and View Designer will not check the data types of the columns you use to create a join, but when you execute the query, the database will display an error if the data types are not compatible.

3. If necessary, change the join operator; by default, the operator is an equal sign (=). For details, see Modify Join Operators (Visual Database Tools).

The Query and View Designer adds an INNER JOIN clause to the SQL statement in the SQL pane. You can change the type to an outer join. For details see Create Outer Joins (Visual Database Tools).

See Also

Query with Joins (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Open Database Diagram Designer (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

The Database Diagram Designer is a visual tool that allows you to design and visualize a database to which you are connected. When designing a database, you can use Database Designer to create, edit, or delete tables, columns, keys, indexes, relationships, and constraints. To visualize a database, you can create one or more diagrams illustrating some or all of the tables, columns, keys, and relationships in it.

You can open Database Diagram Designer by opening a new or existing diagram.

Open a new database diagram

1. Right-click the Database Diagrams node of your database in Object Explorer.
2. From the drop-down menu, click **New Database Diagram**.
3. In the **Add Table** dialog box, choose tables to work with in the diagram.

   > **Note**
   >
   > If you choose **Close** without adding tables, you can go back and add tables later.

4. The **Database Diagram** menu will be added to the main menu and the designer pane will open.

Open an existing database diagram

1. Right-click the diagram under the Database Diagrams node of your database in Object Explorer.
2. From the drop-down menu, click **Design Database Diagram**.
3. The **Database Diagram** menu will be added to the main menu and the diagram will open in the designer pane.

See Also

Design Database Diagrams (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Set Up Database Diagram Designer (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

To use Database Diagram Designer, it must first be set up by a member of the **db_owner** role to control access to diagrams.

To set up database diagramming

1. From Object Explorer, expand a database node.
2. Expand the Database Diagrams node under the database connection.
3. Select **Yes** when prompted if you want to set up database diagramming.

   > **Note**
   >
   > This will create the database diagram table, system stored procedures, and a system function on the SQL Server database.

4. Visual Studio will create the following objects on the instance of SQL Server:

   a. sysdiagrams table

   b. sp_alterdiagram stored procedure

   c. sp_creatediagram stored procedure

   d. sp_dropdiagram stored procedure

   e. sp_renamediagram stored procedure

   f. fn_diagramobjects function

   g. sp_helpdiagrams stored procedure

   h. sp_helpdiagramsdefinition stored procedure

   i. sp_upgraddiagrams stored procedure

See Also

Understand Database Diagram Ownership (Visual Database Tools)
Upgrade Database Diagrams from Previous Editions (Visual Database Tools)
ALTER AUTHORIZATION (Transact-SQL)

Feedback

Was this page helpful? Yes No

Create Self-Joins Manually (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can join a table to itself even if the table does not have a reflexive relationship in the database. For example, you can use a self-join to find pairs of authors living in the same city.

As with any join, a self-join requires at least two tables. The difference is that, instead of adding a second table to the query, you add a second instance of the same table. That way, you can compare a column in the first instance of the table to the same column in the second instance, which allows you to compare the values in a column to each other. The Query and View Designer assigns an alias to the second instance of the table.

For example, if you are creating a self-join to find all pairs of authors within Berkeley, you compare the `city` column in the first instance of the table against the `city` column in the second instance. The resulting query might look like the following:

```
SELECT
         authors.au_fname,
         authors.au_lname,
         authors1.au_fname AS Expr2,
         authors1.au_lname AS Expr3
      FROM
         authors
            INNER JOIN
            authors authors1
               ON authors.city
                = authors1.city
      WHERE
         authors.city = 'Berkeley'
```

Creating a self-join often requires multiple join conditions. To understand why, consider the result of the preceding query:

```
Cheryl Carson       Cheryl Carson
   Abraham Bennet      Abraham Bennet
```

```
   Cheryl Carson       Abraham Bennet
   Abraham Bennet      Cheryl Carson
```

The first row is useless; it indicates that Cheryl Carson lives in the same city as Cheryl Carson. The second row is equally useless. To eliminate this useless data, you add another condition retaining only those result rows in which the two author names describe different authors. The resulting query might look like this:

```
SELECT
         authors.au_fname,
         authors.au_lname,
         authors1.au_fname AS Expr2,
         authors1.au_lname AS Expr3
      FROM
         authors
            INNER JOIN
            authors authors1
               ON authors.city
                = authors1.city
               AND authors.au_id
                <> authors1.au_id
      WHERE
         authors.city = 'Berkeley'
```

The result set is improved:

```
Cheryl Carson       Abraham Bennet
   Abraham Bennet      Cheryl Carson
```

But the two result rows are redundant. The first says Carson lives in the same city as Bennet, and the second says the Bennet lives in the same city as Carson. To eliminate this redundancy, you can alter the second join condition from "not equals" to "less than." The resulting query might look like this:

```
SELECT
         authors.au_fname,
         authors.au_lname,
         authors1.au_fname AS Expr2,
         authors1.au_lname AS Expr3
      FROM
         authors
            INNER JOIN
```

```
            authors authors1
               ON authors.city
                = authors1.city
               AND authors.au_id
                < authors1.au_id
      WHERE
         authors.city = 'Berkeley'
```

And the result set looks like this:

```

Cheryl Carson       Abraham Bennet
```

To create a self-join manually

1. Add to the Diagram pane the table or table-valued object you want to work with.
2. Add the same table again, so that the Diagram pane shows the same table or table-valued object twice within the Diagram pane.

   The Query and View Designer assigns an alias to the second instance by adding a sequential number to the table name. In addition, the Query and View Designer creates a join line between the two occurrences of the table or table-valued object within the Diagram pane.
3. Right-click the join line and choose **Properties** from the shortcut menu.
4. In the Properties window click **Join Condition and Type** and click the **ellipses (...)** to the right of the property.
5. In the Join dialog box, change the comparison operator between the primary keys as required. For example, you might change the operator to less than (<).
6. Create the additional join condition (for example, authors.zip = authors1.zip) by dragging the name of the primary join column in the first occurrence of the table or table-valued object and dropping it on the corresponding column in the second occurrence.
7. Specify other options for the query such as output columns, search conditions, and sort order.

See Also

Create Self-Joins Automatically (Visual Database Tools)

Query with Joins (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Remove Tables from Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can remove a table - or any table-valued object - from the query.

> **Note**
>
> Removing a table or table-valued object does not delete anything from the database; it only removes it from the current query. For details about removing a table from a database, see How to: Delete Tables from a Database.

To remove a table or table-structured object

- In the **Diagram Pane**, select the table, view, user-defined function, synonym, or query, and then press DELETE, or right-click the object and then choose **Remove** in the resulting dialog box. You can select and remove multiple objects at one time.

  -or-

- Remove all references to the object in the **SQL Pane.**

When you remove a table or table-valued object, the Query and View Designer automatically removes joins that involve that table or table-valued object and removes references to the object's columns in the **SQL Pane** and **Criteria Pane**. However, if the query contains complex expressions involving the object, the object is not automatically removed until all references to it are removed.

See Also

Add Tables to Queries
Create Table Aliases
Specify Search Criteria
Summarize Query Results
Perform Basic Operations with Queries

Feedback

Was this page helpful? Yes No

Query with Joins (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

A query result can include data from multiple tables or table-valued objects. To combine data from multiple table-valued objects, you use the JOIN operation from SQL.

For information about creating queries using multiple tables, see the following topics.

In This Section

Modify Join Operators (Visual Database Tools)
Specify that tables should be joined using an operator other than equal (=).

How the Query and View Designer Represents Joins (Visual Database Tools)
Explains the graphic representation of the join as you see it in the Diagram pane.

Join Tables Automatically (Visual Database Tools)
Steps for allowing the Query and View Designer determine if tables should be joined.

Join Tables Manually (Visual Database Tools)
Steps for creating a join manually in the Diagram pane.

Join Tables on Multiple Columns (Visual Database Tools)
Steps for joining tables with multiple criteria for each table.

Create Outer Joins (Visual Database Tools)
Specify that joined tables should include rows even when they do not match rows in the corresponding table.

Remove Joins (Visual Database Tools)
Steps for removing a join between tables.

Create Self-Joins Automatically (Visual Database Tools)
Steps for allowing the Query and View Designer to create a self-join.

Create Self-Joins Manually (Visual Database Tools)
Steps for using a join to find subsets of data within a single table.

View Join Properties (Visual Database Tools)
Steps for viewing the properties of a join.

Related Sections

Types of Queries (Visual Database Tools)
Provides links to topics describing the supported query types.

Perform Basic Operations with Queries (Visual Database Tools)
Provides links to topics covering the most common query tasks.

Specify Search Criteria (Visual Database Tools)
Provides links to topics covering the various kinds of search criteria and how to use them.

Feedback

Was this page helpful? Yes No

Create Queries using Something Besides a Table (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

Whenever you write a retrieval query, you articulate what columns you want, what rows you want, and where the query processor should find the original data. Typically, this original data consists of a table or several tables joined together. But the original data can come from sources other than tables. In fact, it can come from views, queries, synonyms, or user-defined functions that return a table.

Using a View in Place of a Table

You can select rows from a view. For example, suppose the database includes a view called "ExpensiveBooks," in which each row describes a title whose price exceeds 19.99. The view definition might look like this:

```
SELECT *
FROM titles
WHERE price > 19.99
```

You can select the expensive psychology books merely by selecting the psychology books from the ExpensiveBooks view. The resulting SQL might look like this:

```
SELECT *
FROM ExpensiveBooks
WHERE type = 'psychology'
```

Similarly, a view can participate in a JOIN operation. For example, you can find the sales of expensive books merely by joining the sales table to the ExpensiveBooks view. The resulting SQL might look like this:

```
SELECT *
FROM sales
         INNER JOIN
```

```
        ExpensiveBooks
        ON sales.title_id
        =  ExpensiveBooks.title_id
```

For more information about adding a view to a query, see Add Tables to Queries (Visual Database Tools).

Using a Query in Place of a Table

You can select rows from a query. For example, suppose you have already written a query retrieving titles and identifiers of the coauthored books - the books with more than one author. The SQL might look like this:

```
SELECT
     titles.title_id, title, type
FROM
     titleauthor
         INNER JOIN
         titles
         ON titleauthor.title_id
         =  titles.title_id
GROUP BY
     titles.title_id, title, type
HAVING COUNT(*) > 1
```

You can then write another query that builds on this result. For example, you can write a query that retrieves the coauthored psychology books. To write this new query, you can use the existing query as the source of the new query's data. The resulting SQL might look like this:

```
SELECT
    title
FROM
    (
    SELECT
        titles.title_id,
        title,
        type
    FROM
        titleauthor
            INNER JOIN
            titles
            ON titleauthor.title_id
            =  titles.title_id
```

```
    GROUP BY
        titles.title_id,
        title,
        type
    HAVING COUNT(*) > 1
    )
    co_authored_books
WHERE     type = 'psychology'
```

The emphasized text shows the existing query used as the source of the new query's data. Note that the new query uses an alias ("co_authored_books") for the existing query. For more information about aliases, see Create Table Aliases (Visual Database Tools) and Create Column Aliases (Visual Database Tools).

Similarly, a query can participate in a JOIN operation. For example, you can find the sales of expensive coauthored books merely by joining the ExpensiveBooks view to the query retrieving the coauthored books. The resulting SQL might look like this:

```
SELECT
    ExpensiveBooks.title
FROM
    ExpensiveBooks
        INNER JOIN
        (
        SELECT
            titles.title_id,
            title,
            type
        FROM
            titleauthor
                INNER JOIN
                titles
                ON titleauthor.title_id
                =  titles.title_id
        GROUP BY
            titles.title_id,
            title,
            type
        HAVING COUNT(*) > 1
        )
```

For more information about adding a query to a query, see Add Tables to Queries (Visual Database Tools).

Using a User-Defined Function in Place of a Table

In SQL Server 2000 or higher, you can create a user-defined function that returns a table. Such functions are useful for performing complex or procedural logic.

For example, suppose the employee table contains an additional column, employee.manager_emp_id, and that a foreign key exists from manager_emp_id to employee.emp_id. Within each row of the employee table, the manager_emp_id column indicates the employee's boss. More precisely, it indicates the employee's boss's emp_id. You can create a user-defined function that returns a table containing one row for each employee working within a particular high-level manager's organizational hierarchy. You might call the function fn_GetWholeTeam, and design it to take an input variable - the emp_id of the manager whose team you want to retrieve.

You can write a query that uses the fn_GetWholeTeam function as a source of data. The resulting SQL might look like this:

```
SELECT *
FROM
     fn_GetWholeTeam ('VPA30890F')
```

"VPA30890F" is the emp_id of the manager whose organization you want to retrieve. For more information about adding a user-defined function to a query, see Add Tables to Queries (Visual Database Tools). For a complete description of user-defined functions, see User-Defined Functions.

Feedback

Was this page helpful? Yes No

Perform Basic Operations with Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

The following fundamental operations apply to most or all queries.

In This Section

Create Queries (Visual Database Tools)
Provides steps for creating new queries.

Open Queries (Visual Database Tools)
Provides steps for opening existing queries.

Run Queries (Visual Database Tools)
Provides steps for running (or executing) queries.

Stop a Query (Visual Database Tools)
Provides steps for halting a query while it's running.

Save Queries (Visual Database Tools)
Provides steps for saving new or revised queries to a database project.

Verify Queries (Visual Database Tools)
Provides steps for verifying the SQL syntax of a query.

Modify Queries (Visual Database Tools)
Provides steps for opening an existing query for modification.

Delete Queries (Visual Database Tools)
Provides steps for deleting queries from projects.

Copy Queries (Visual Database Tools)
Provides steps for copying the SQL code form one query and pasting it into another.

Rename Queries (Visual Database Tools)
Provides steps for renaming existing queries

Discard Changes Made to Queries (Visual Database Tools)
Provides steps for reverting a query to its last changed state.

Add Tables to Queries (Visual Database Tools)
Provides steps for adding tables and other table-valued objects, such as table-valued functions and views, to queries.

Remove Tables from Queries (Visual Database Tools)
Provides steps for removing tables and other table-valued objects from the structure of a query.

Add Columns to Queries (Visual Database Tools)
Provides steps for adding columns to queries.

Remove Columns from Queries (Visual Database Tools)
Provides steps for removing columns from the structure of a query.

Create Column Aliases (Visual Database Tools)
Provides steps for creating aliases for columns included in a query.

Create Table Aliases (Visual Database Tools)
Provides steps for creating aliases for tables included in a query.

Show Query Properties (Visual Database Tools)
Provides steps for showing properties such as SQL comments, parameter lists, and Top specification.

Create Queries using Something Besides a Table (Visual Database Tools)
Provides steps for creating queries that use table-valued objects, such as table-valued functions and views.

Show Column Names in the Diagram Pane (Visual Database Tools)
Provides steps for customizing how tables and table-valued objects look in the Diagram pane.

Use the Query and View Designer with International Data (Visual Database Tools)
Describes the features the Query and View Designer provide to make working with international data easier.

Reference

Query and View Designer Tools (Visual Database Tools)
Describes the features and capabilities of the Query and View Designer.

Query Properties (Visual Database Tools)
Describes the properties available for queries.

Related Sections

Types of Queries (Visual Database Tools)
Lists topics concerning queries you can work with in Query and View Designer.

Specify Search Criteria (Visual Database Tools)
Lists topics about how to create queries that return the results set you want.

Sort and Group Query Results (Visual Database Tools)
Lists topics about how to create queries that return the results set in the organization you want.

Summarize Query Results (Visual Database Tools)
Lists topics about how to create queries that summarize results of the involved tables and table-valued objects.

Design Queries and Views How-to Topics (Visual Database Tools)
Lists overview topics that provide more detailed links.

Feedback

Was this page helpful? Yes No

Create Insert Results Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can copy rows from one table to another or within a table using an Insert Results query. For example, in a `titles` table, you can use an Insert Results query to copy information about all the titles for one publisher to a second table that you can make available to that publisher. An Insert Results query is similar to Make Table Queries, but copies rows into an existing table.

> **Tip**
>
> You can also copy rows from one table to another using cut and paste. Create a query for each table and run the queries. Copy the rows you want from one results grid to the other.

When you create an Insert Results query, you specify:

- The database table to copy rows to (the destination table).
- The table or tables to copy rows from (the source table). The source table or tables become part of a subquery. If you are copying within a table, the source table is the same as the destination table.
- The columns in the source table whose contents you want to copy.
- The target columns in the destination table to copy the data to.
- Search conditions to define the rows you want to copy.
- Sort order, if you want to copy the rows in a particular order.
- Group By options, if you want to copy only summary information.

For example, the following query copies title information from the `titles` table to an archive table called `archivetitles`. The query copies the contents of four columns for all titles belonging to a particular publisher:

```
```

```
INSERT INTO archivetitles
   (title_id, title, type, pub_id)
SELECT title_id, title, type, pub_id
FROM titles
WHERE (pub_id = '0766')
```

> ⓘ **Note**
>
> To insert values into a new row, use an Insert Values query.

You can copy the contents of selected columns or of all columns in a row. In either case, the data you are copying must be compatible with the columns in the rows you are copying to. For example, if you copy the contents of a column such as `price`, the column in the row you are copying to must accept numeric data with decimal places. If you are copying an entire row, the destination table must have compatible columns in the same physical position as the source table.

When you create an Insert Results query, the Criteria pane changes to reflect options available for copying data. An Append column is added to allow you to specify the columns into which data should be copied.

> ⊗ **Caution**
>
> You cannot undo the action of executing an Insert Results query. As a precaution, back up your data before executing the query.

To create an Insert Results query

1. Create a new query and add the table from which you want to copy rows (the source table). If you are copying rows within a table, you can add the source table as a destination table.
2. From the **Query Designer** menu, point to **Change Type**, and then click **Insert Results**.
3. In the Choose Target Table for Insert Results dialog box, select the table to copy rows to (the destination table).

   > ⓘ **Note**

> The Query and View Designer cannot determine in advance which tables and views you can update. Therefore, the **Table Name** list in the **Choose Table for Insert From Query** dialog box shows all available tables and views in the data connection you are querying, even those that you might not be able to copy rows to.

4. In the rectangle representing the table or table-valued object, choose the names of the columns whose contents you want to copy. To copy entire rows, choose *** (All Columns)**.

   The Query and View Designer adds the columns you choose to the **Column** column of the Criteria pane.

5. In the **Append** column of the Criteria pane, select a target column in the destination table for each column you are copying. Choose *tablename*.* if you are copying entire rows. The columns in the destination table must have the same (or compatible) data types as the columns in the source table.

6. If you want to copy rows in a particular order, specify a sort order. For details, see Sort and Group Query Results (Visual Database Tools).

7. Specify the rows to copy by entering search conditions in the **Filter** column. For details, see Specify Search Criteria (Visual Database Tools).

   If you do not specify a search condition, all rows from the source table will be copied to the destination table.

   > **Note**
   >
   > When you add a column to search to the Criteria pane, the Query and View Designer also adds it to the list of columns to copy. If you want to use a column for searching but not copy it, clear the check box next to the column name in the rectangle representing the table or table-valued object.

8. If you want to copy summary information, specify Group By options. For details, see Summarize Query Results (Visual Database Tools).

When you execute an Insert Results query, no results are reported in the Results Pane. Instead, a message appears indicating how many rows were copied.

See Also

Types of Queries (Visual Database Tools)

Design Queries and Views How-to Topics (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Save Selected Tables on a Diagram (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

You can save a specific table or a set of tables if you do not want to save all the changes you made in a database diagram.

To save selected tables

1. In your database diagram, select the tables you want to save.
2. From the **File** menu, choose **Save Selection**.
3. The **Save** dialog box displays the list of tables that will be updated in the database when you save your selection.

   Choose **Save Text File** if you want to save the list of tables in a text file in the project directory before continuing.
4. In the **Save** dialog box, confirm the list of tables and choose **Yes** to save these tables.

> **Note**
>
> The list of tables may contain tables in addition to those selected. For example, if you change the data type of a column that participates in a relationship with another table, both tables will be included in this list.

See Also

Work with Database Diagrams

Feedback

Was this page helpful? Yes No

Show Diagram Properties (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

You can display properties for a database diagram in the Properties window.

To show database diagram properties in the Properties window

1. Open Database Diagram Designer.
2. Make sure that no objects are selected in Database Diagram Designer by clicking anywhere within the designer except on an object.
3. From the **View** menu, click **Properties Window**.

The properties for the diagram are now visible in the **Properties** window.

See Also

Work with Database Diagrams (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Create a New Database Diagram (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

You can use Object Explorer to create new database diagrams. Database diagrams graphically show the structure of the database. Using database diagrams you can create and modify tables, columns, relationships, and keys. Additionally, you can modify indexes and constraints.

To create a new database diagram

1. In Object Explorer, right-click the **Database Diagrams** folder or any diagram in that folder.
2. Choose **New Database Diagram** on the shortcut menu.

   The **Add Table** dialog box appears.
3. Select the required tables in the **Tables** list and click **Add**.

   The tables are displayed graphically in the new database diagram.

You can continue to add or delete tables, modify the existing tables, and alter table relationships until the new database diagram is complete.

See Also

Work with Database Diagrams (Visual Database Tools)
Understand Database Diagram Ownership (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Add Columns to Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

To use a column in a query, you must add it to the query. You might add a column to display it in query output, to use it for sorting, to search the contents of the column, or to summarize its contents. You can decide which of the columns you use in the query are included in the results pane when the query is run. For more information see Remove Columns from Query Results (Visual Database Tools).

> **Note**
>
> To view the data type of a column in Query and View Designer; select the table or table-valued object in the **Diagram Pane** and in the properties window click Column List. Then click the **ellipses (...)** to open the **Column List** dialog box.

Wherever you use a column in a query, you can also use an expression that can consist of any combination of columns, literals, operators, and functions.

To add an individual column

- In the **Diagram Pane**, select the check box next to the column that you want to include.

  -or-

- In the **Criteria Pane**, move to the first blank grid row, click the field in the **Column** column, and select a column name from the drop-down list.

To add all columns for one table or table-valued object

- In the **Diagram Pane**, select the check box next to ***(All Columns)**.

To add all columns for all tables and table-structured objects

1. Make sure no join lines in the **Table Operations Pane** are selected.

2. Right-click in the empty space of the Design window and choose **Properties** from the shortcut menu.

3. In the Properties window click **Output all columns** and choose **Yes** or **No** from the dropdown list.

See Also

Remove Columns from Query Results (Visual Database Tools)
Remove Columns from Queries (Visual Database Tools)
Specify Search Criteria (Visual Database Tools)
Summarize Query Results (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Copy an Image of a Database Diagram to the Clipboard (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server, Azure SQL Database, Azure SQL Managed Instance, Analytics Platform System (PDW)

You can create an image of a Database Diagram and copy it to another application.

To copy a diagram image to the Clipboard

1. Open a Database Diagram.

   > **Note**
   >
   > Only the owner of the diagram or a member of the db_owner role of the database can open the diagram.

2. Right-click a blank area and choose **Copy Diagram to Clipboard**.

   The image of the entire Database Diagram is now in the system Clipboard.

See Also

Work with Database Diagrams (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Discard Changes Made to Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can discard changes made to a query definition prior to saving them. Once they have been saved, they cannot be returned to a previous state.

> **Note**
>
> To undo a change you've made to values in the Results pane, press the ESC key before you move off of the record. If you move off of a record and receive a notice that the changes will not be committed to the database you can also press the ESC key to revert to the previous value.

To discard changes made to a query definition

1. With the query open in Query and View Designer, from the **File** menu, click **Close**.
2. In the **Microsoft SQL Server Management Studio** dialog box, click **No**.

   The query definition will return to the state it was in at the last save.

See Also

Save Queries
Design Queries and Views How-to Topics
Perform Basic Operations with Queries
Work with Data in the Results Pane

Feedback

Was this page helpful? Yes No

Visual Database Tools F1 Help

Article • 02/13/2025

Applies to: SQL Server

These articles discuss the commands and options in the dialog boxes and properties windows for Visual Database Tools elements such as the Table Properties window, and the Column Properties tab.

To access dialog box articles, press F1 when you have a dialog box open and it has focus. To access properties window articles, select in the Properties window and press F1.

Related content

Design Tables (Visual Database Tools)
Design Queries and Views How-to articles (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Sort Rows (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can order the rows in a query result. That is, you can name a particular column or set of columns whose values determine the order of rows in the result set.

> **Note**
>
> The sort order is determined in part by the column's collation sequence. You can change the collation sequence in the Collation dialog box.

There are several ways in which you can sort query results:

- **You can arrange rows in ascending or descending order** By default, SQL uses order-by columns to arrange rows in ascending order. For example, to arrange the book titles by ascending price, simply sort the rows by the price column. The resulting SQL might look like this:

  ```
  SELECT *
  FROM titles
  ORDER BY price
  ```

  On the other hand, if you want to arrange the titles with the more expensive books first, you can explicitly specify a highest-first ordering. That is, you indicate that the result rows should be arranged by descending values of the price column. The resulting SQL might look like this:

  ```
  SELECT *
  FROM titles
  ORDER BY price DESC
  ```

- **You can sort by multiple columns** For example, you can create a result set with one row for each author, ordering first by state and then by city. The resulting SQL might look like this:

```
SELECT *
FROM authors
ORDER BY state, city
```

- **You can sort by columns not appearing in the result set** For example, you can create a result set with the most expensive titles first, even though the prices do not appear. The resulting SQL might look like this:

```

SELECT title_id, title
FROM titles
ORDER BY price DESC
```

- **You can sort by derived columns** For example, you can create a result set in which each row contains a book title - with the books that pay the highest royalty per copy appearing first. The resulting SQL might look like this:

```

SELECT title, price * royalty / 100 as royalty_per_unit
FROM titles
ORDER BY royalty_per_unit DESC
```

  (The formula for calculating the royalty that each book earns per copy is emphasized.)

  To calculate a derived column, you can use SQL syntax, as in the preceding example, or you can use a user-defined function that returns a scalar value. For more information about user-defined functions, see the SQL Server documentation.

- **You can sort grouped rows** For example; you can create a result set in which each row describes a city, plus the number of authors in that city - with the cities containing many authors appearing first. The resulting SQL might look like this:

```

SELECT city, state, COUNT(*)
FROM authors
GROUP BY city, state
ORDER BY COUNT(*) DESC, state
```

Notice that the query uses `state` as a secondary sort column. Thus, if two states have the same number of authors, those states will appear in alphabetical order.

- **You can sort using international data** That is; you can sort a column using collating conventions that differ from the default conventions for that column. For example, you can write a query that retrieves all the book titles by Jaime Patiño. To display the titles in alphabetical order, you use a Spanish collating sequence for the title column. The resulting SQL might look like this:

```
SELECT title
FROM
    authors
    INNER JOIN
        titleauthor
        ON authors.au_id
        =  titleauthor.au_id
        INNER JOIN
            titles
            ON titleauthor.title_id
            =  titles.title_id
WHERE
     au_fname = 'Jaime' AND
     au_lname = 'Patiño'
ORDER BY
     title COLLATE SQL_Spanish_Pref_CP1_CI_AS
```

See Also

Sort and Group Query Results (Visual Database Tools)
Design Queries and Views How-to Topics (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Types of Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

The Query and View Designer provides built-in features for some query types--Select, Insert Results, Insert Values, Update, Delete, and Make Table--and also supports UNION, Full-text search, and CASE expression queries.

In This Section

Supported Query Types (Visual Database Tools)
Describes the query types you can use with the Query and View Designer.

Create Full-Text Search Queries (Visual Database Tools)
Describes how to create a query that searches a full-text indexed column.

Create Insert Results Queries (Visual Database Tools)
Describes how to create a query that inserts the results of a SELECT statement into a table.

Create Insert Values Queries (Visual Database Tools)
Describes how to create a query that inserts values into a table.

Create Make Table Queries (Visual Database Tools)
Describes how to create a query that creates a new table.

Create Delete Queries (Visual Database Tools)
Describes how to create a query that deletes data.

Create UNION Queries (Visual Database Tools)
Describes how to create a query that merges the results of two or more SELECT statements into one result set.

Create Update Queries (Visual Database Tools)
Describes how to create a query that updates a table.

Reference

Query Properties (Visual Database Tools)
Describes the properties available to you when designing queries.

Related Sections

Design Queries and Views How-to Topics (Visual Database Tools)
Describes how to design and manipulate queries and views.

Feedback

Was this page helpful? Yes No

Reorder Output Columns (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

The order in which you add data columns to a Select query determines the order in which they appear in the results. The first column you add to the query appears leftmost in the results, the second column next, and so on.

If you are creating Update or Insert queries, the order in which you add columns affects the order in which data is processed.

To control where a data column appears in the result set, or in what order it is used, you can reorder the columns.

To reorder columns for output

1. In the Criteria pane, select the row containing the column by clicking the row selector button to the left of the row.
2. Point to the row selector button and drag the row to a new location.

   -or-

   Edit the order of the column names in the SQL pane.

> **Tip**
>
> You can add a data row at a specific location in the Criteria pane by inserting a blank row into the Criteria pane, and then specifying the data column to insert. For details, see **Add Columns to Queries (Visual Database Tools)**.

See Also

Sort and Group Query Results (Visual Database Tools)
Summarize Query Results (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Save Database Diagrams (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

By saving a database diagram, you can save all the changes you made to it, including any changes you made to the tables, columns, and other database objects.

To save the database diagram

1. From the **File** menu, choose **Save** <*tablename*>.
2. If this is a new diagram that you have never before saved, the **Save As** dialog box appears. Type a name for the diagram.

   If you made changes to tables in an existing diagram, the **Save** dialog box appears and displays a list of changes that will be saved in the database when you save your diagram.
3. Click **Yes** (or **OK** if this is a new diagram) to update the database to match your diagram.

See Also

Work with Database Diagrams (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Collapse Groups of Rows (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can create a query result in which each result row corresponds to an entire group of rows from the original data. When collapsing rows, there are several things to keep in mind:

- **You can eliminate duplicate rows** Some queries can create result sets in which multiple identical rows appear. For example, you can create a result set in which each row contains the city and state name of a city containing an author - but if a city contains several authors, there will be several identical rows. The resulting SQL might look like this:

  ```
  SELECT city, state
  FROM authors
  ```

  The result set generated by the preceding query is not very useful. If a city contains four authors, the result set will include four identical rows. Since the result set does not include any columns other than city and state, there is no way to distinguish the identical rows from each other. One way to avoid such duplicate rows is to include additional columns that can make the rows different. For example, if you include author name, each row will be different (provided no two like-named authors live within any one city). The resulting SQL might look like this:

  ```
  SELECT city, state, fname, minit, lname
  FROM authors
  ```

  Of course, the preceding query eliminates the symptom, but does not really solve the problem. That is, the result set has no duplicates, but it is no longer a result set about cities. To eliminate duplicates in the original result set, and still have each row describe a city, you can create a query returning only distinct rows. The resulting SQL might look like this:

  ```
  ```

```
SELECT DISTINCT city, state
FROM authors
```

For details about eliminating duplicates, see Exclude Duplicate Rows (Visual Database Tools).

- **You can calculate on groups of rows** That is, you can summarize information in groups of rows. For example, you can create a result set in which each row contains the city and state name of a city containing an author, plus a count of the number of authors contained in that city. The resulting SQL might look like this:

```
SELECT city, state, COUNT(*)
FROM authors
GROUP BY city, state
```

For details about calculating on groups of rows, see Summarize Query Results (Visual Database Tools) and Sort and Group Query Results (Visual Database Tools).

- **You can use selection criteria to include groups of rows** For example, you can create a result set in which each row contains the city and state name of a city containing several authors, plus a count of the number of authors contained in that city. The resulting SQL might look like this:

```
SELECT city, state, COUNT(*)
FROM authors
GROUP BY city, state
HAVING COUNT(*) > 1
```

For details about applying selection criteria on groups of rows, see Specify Conditions for Groups (Visual Database Tools) and Use HAVING and WHERE Clauses in the Same Query (Visual Database Tools).

See Also

Specify Search Criteria (Visual Database Tools)
Design Queries and Views How-to Topics (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Supported Query Types (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can create the following types of queries in the Diagram and Criteria panes (the graphical panes) of the Query and View Designer:

- **Select query** Retrieves data from one or more tables or views. This type of query creates a SQL SELECT statement.
- **Insert Results** Creates new rows by copying existing rows from one table into another, or into the same table as new rows. This type of query creates a SQL INSERT INTO...SELECT statement.
- **Insert Values** Creates a new row and inserts values into specified columns. This type of query creates a SQL INSERT INTO...VALUES statement.
- **Update query** Changes the values of individual columns in one or more existing rows in a table. This type of query creates a SQL UPDATE...SET statement.
- **Delete query** Removes one or more rows from a table. This type of query creates a SQL DELETE statement.

  > **Note**
  >
  > A Delete query removes entire rows from the table. If you want to delete values from individual data columns, use an Update query.

- **Make Table query** Creates a new table and creates rows in it by copying the results of a query into it. This type of query creates a SQL SELECT...INTO statement.

In addition to the queries you can create using the graphical panes, you can enter any SQL statement into the SQL pane, such as Union queries.

When you create queries using SQL statements that cannot be represented in the graphical panes, the Query and View Designer dims those panes to indicate that they do not reflect the query you are creating. However, the dimmed panes are still active and, in many cases, you can make changes to the query in those panes. If the changes you make result in a query that can be represented in the graphical panes, those panes are no longer dimmed.

See Also

Design Queries and Views How-to Topics
Types of Queries

Feedback

Was this page helpful? Yes No

Remove Columns from Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

If you no longer want to use a column in a query, you can remove it. If you do, the Query and View Designer removes references to the column in the select list, the sort specification, the search criteria, **SQL Pane**, and any grouping specifications.

> **Note**
>
> If you want to remove a column from just the output of a Select query, you can do so without removing it from the query altogether. For details, see Remove Columns from Query Results (Visual Database Tools).

To remove a column from the query

- In the **Criteria Pane**, select the grid row containing the column you want to remove and then press DELETE.

  -or-

- Remove all references to the column in the SQL Pane.

See Also

Add Columns to Queries (Visual Database Tools)
Sort and Group Query Results (Visual Database Tools)
Summarize Query Results (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Open Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can open an existing query in Query and View Designer by using the **Open** command.

To open a query in all panes

1. From the **View** menu, click **Solution Explorer**.
2. In Solution Explorer, right-click the query you want to open and click **Open**.
3. To modify the query in Query Designer, highlight the SQL statements, right-click the highlighted area and select **Design Query in Editor.**

> **Note**
>
> By default the Query and View Designer opens all of its panes (SQL, Criteria, Diagram, and Results), but that default can be changed in the **Options** dialog box. To open these panes if they are closed, from the **Query Designer** menu, point to **Pane**, and then click the pane you want to open.

See Also

Perform Basic Operations with Queries (Visual Database Tools)
Save Queries (Visual Database Tools)
Work with Data in the Results Pane (Visual Database Tools)

Feedback

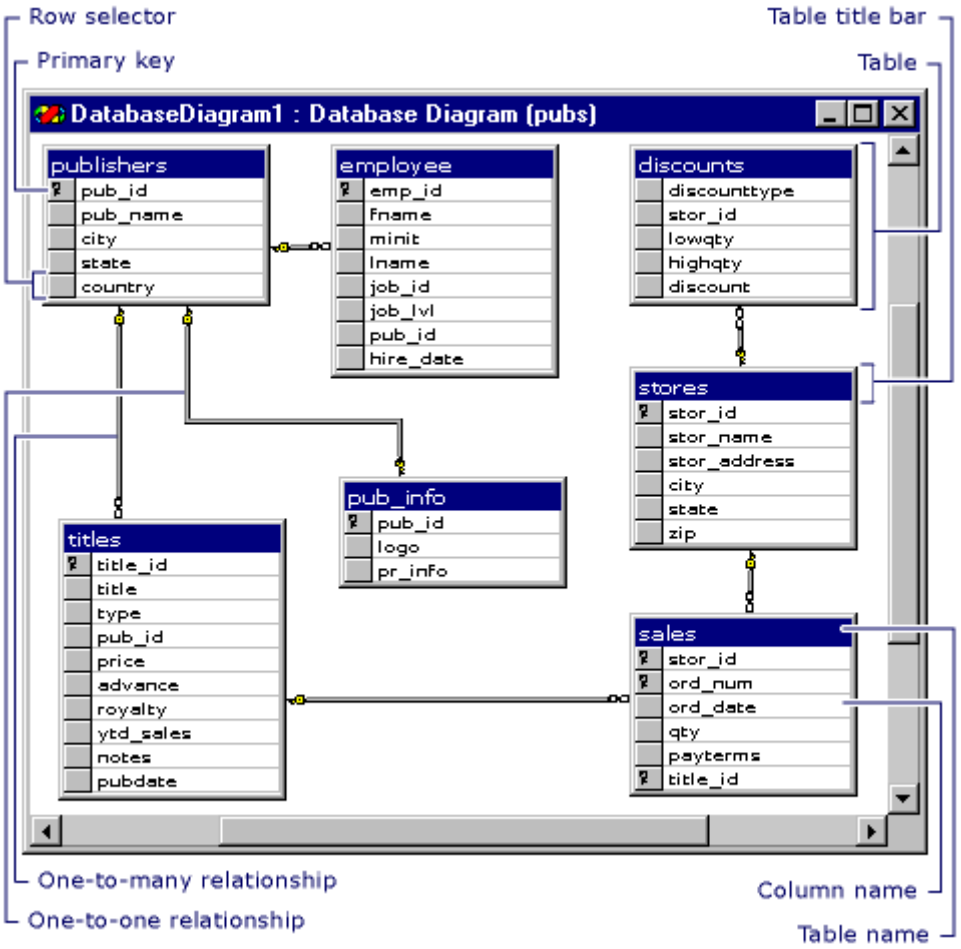
Was this page helpful? Yes No

Design Database Diagrams (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

The Database Designer is a visual tool that allows you to design and visualize a database to which you are connected. When designing a database, you can use Database Designer to create, edit, or delete tables, columns, keys, indexes, relationships, and constraints. To visualize a database, you can create one or more diagrams illustrating some or all of the tables, columns, keys, and relationships in it.



For any database, you can create as many database diagrams as you like; each database table can appear on any number of diagrams. Thus, you can create different diagrams to visualize different portions of the database, or to accentuate different aspects of the design. For example, you can create a large diagram showing all tables and columns, and you can create a smaller diagram showing all tables without showing the columns.

Each database diagram you create is stored in the associated database.

Tables and Columns in a Database Diagram

Within a database diagram, each table can appear with three distinct features: a title bar, a row selector, and a set of property columns.

Title Bar The title bar shows the name of the table

If you have modified a table and have not yet saved it, an asterisk (*) appears at the end of the table name to indicate unsaved changes. For information about saving modified tables and diagrams, see Work with Database Diagrams (Visual Database Tools)

Row Selector You can click the row selector to select a database column in the table. The row selector displays a key symbol if the column is in the table's primary key. For information about primary keys, see Working with Keys.

Property Columns The set of property columns is visible only in the certain views of your table. You can view a table in any of five different views to help you manage the size and layout of your diagram.

For more information about table views, see Customize the Amount of Information Displayed in Diagrams (Visual Database Tools).

Relationships in a Database Diagram

Within a database diagram, each relationship can appear with three distinct features: endpoints, a line style, and related tables.

Endpoints The endpoints of the line indicate whether the relationship is one-to-one or one-to-many. If a relationship has a key at one endpoint and a figure-eight at the other, it is a one-to-many relationship. If a relationship has a key at each endpoint, it is a one-to-one relationship.

Line Style The line itself (not its endpoints) indicates whether the Database Management System (DBMS) enforces referential integrity for the relationship when new data is added to the foreign-key table. If the line appears solid, the DBMS enforces referential integrity for the relationship when rows are added or modified in the foreign-key table. If the line appears dotted, the DBMS does not enforce referential integrity for the relationship when rows are added or modified in the foreign-key table.

Related Tables The relationship line indicates that a foreign-key relationship exists between one table and another. For a one-to-many relationship, the foreign-key table is the table near the line's figure-eight symbol. If both endpoints of the line attach to the

same table, the relationship is a reflexive relationship. For more information, see Draw Reflexive Relationships (Visual Database Tools).

In this Section

Understand Database Diagram Ownership (Visual Database Tools)

Navigate in Database Diagram Designer (Visual Database Tools)

Set Up Database Diagram Designer (Visual Database Tools)

Upgrade Database Diagrams from Previous Editions (Visual Database Tools)

Open Database Diagram Designer (Visual Database Tools)

See Also

Work with Database Diagrams (Visual Database Tools)
Work with Tables in Database Diagram (Visual Database Tools)
Work with Diagram Layout (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Run Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

When you have finished designing your query, you can run it.

To execute a query

1. Open or create the query you want to run.
2. Right-click anywhere in the query window, and select **Execute SQL** from the shortcut menu.

   -or-

   Press CTRL+R.

If you are creating a Select query, the results of the query appear in the Results pane.

The Query and View Designer returns results to your computer in batches (incrementally) so that you can begin viewing results as soon as possible, and so that you can perform other tasks while the query is underway.

If you are creating an Update, Insert From, Insert Into, Delete, or Make Table query, the Query and View Designer displays a message indicating how many rows were affected by the query.

See Also

Work with Data in the Results Pane (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Create Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

Queries allow you to retrieve data from the tables and views in your database. You create and work with queries in **Query and View Designer**, which is composed of four panes: the Diagram Pane, the SQL Pane, the Criteria Pane, and the Results Pane.

To create a new query

1. In **Object Explorer**, expand the **Tables** node for the database you want to query. Right-click the table you want to query and click **Open Table**.
2. To add more tables to the query, on the Query Designer menu, select **Add Table**.

   > **Note**
   >
   > If you do not see the **Diagram**, **SQL**, **Criteria**, or **Results** panes, from the Query Designer menu, point to **Pane** and click the pane you want to open.

3. In the **Add Table** dialog box, select the tables you want to query and click **Add** for each one.
4. Once you have added all the tables you want to query, click **Close**.

   To add more tables later, right-click the open space in the **Diagram** pane and from the shortcut menu click **Add Table**.

5. In the **Diagram Pane**, check the boxes in the table-valued objects for each column you want to query.
6. From the Query Designer menu, choose **Execute SQL** to run your query.

To further refine your query, you can change the SQL code in the **SQL Pane** or choose options such as sort order and column aliases in the **Criteria Pane.**

See Also

Save Queries (Visual Database Tools)
Types of Queries (Visual Database Tools)
Specify Search Criteria (Visual Database Tools)

Summarize Query Results (Visual Database Tools)

Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Copy Query Results (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can copy the result set as tab-separated text from the **Results** pane. You may copy the entire result set or any number of contiguous or non-contiguous rows or cells.

To copy result-set text from the Results pane

1. Run the query or view that has the data you want to copy.
2. To select the entire result set, in the **Results** pane, click the top-left empty cell.
3. To select rows:
   - Select one row by clicking the box to the left.
   - Select multiple contiguous rows by selecting one row and then dragging the mouse to select additional rows.
   - Select non-contiguous rows by selecting one row, then holding down the CTRL key and clicking additional rows.

   The selected rows will be highlighted.
4. To select cells:
   - Select a single cell by clicking the value.
   - Select multiple cells by dragging the mouse to adjacent cells.
   - Select non-contiguous cells by selecting a single cell, then holding down the CTRL key and clicking additional cells.

   The selected cells will be highlighted.
5. Right-click and choose **Copy**.

See Also

Work with Data in the Results Pane (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Add Tables to Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

When you create a query, you are retrieving data from a table or other objects structured like tables - views and certain user-defined functions. To work with any of these objects in your query, you add them to the **Diagram Pane**.

To add a table or table-valued object to a query

1. In the **Diagram pane** of the Query and View Designer, right-click the background and choose **Add Table** from the shortcut menu.
2. In the **Add Table** dialog box, select the tab for the type of object you want to add to the query.
3. In the list of items, double-click each item you want to add.
4. When you finish adding items, click **Close**.

   The Query and View Designer updates the **Diagram Pane**, **Criteria Pane**, and **SQL Pane** accordingly.

Tables and views are automatically added to the query when you reference them in the statement in the SQL pane.

The Query and View Designer will not display data columns for a table or table-structured object if you do not have sufficient access rights to it or if the provider cannot return information about it. In such cases, only a title bar and the * (All Columns) check box are displayed for the table or table-valued object.

To add an existing query to a new query

1. Make sure the **SQL Pane** is displayed in the new query you are creating.
2. In the **SQL Pane**, type a right and left parentheses () after the word FROM.
3. Open the Query Designer for the existing query. (You now have two Query Designers open.)

4. Display the **SQL Pane** for the inner query - the existing query you are including in the new, outer query.

5. Select all the text in the **SQL Pane**, and copy it to the Clipboard.

6. Click in the **SQL Pane** of the new query, situate the cursor between the parentheses you added, and paste the contents of the Clipboard.

7. Still in the **SQL Pane**, add an alias after the right parenthesis.

See Also

Create Table Aliases (Visual Database Tools)
Remove Tables from Queries (Visual Database Tools)
Specify Search Criteria (Visual Database Tools)
Summarize Query Results (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Results Pane (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

The Results pane shows the results of the most recently executed SELECT query. (The results of other query types are displayed in message boxes.) To open the results pane, open or create a query or view or return a table's data. If the results pane doesn't show by default, from the **Query Designer** menu, point to **Pane**, and then click **Results**.

What You Can Do in the Results Pane

- View the result set for the most recently executed SELECT query in a spreadsheet-like grid.
- For queries or views that show data from a single table or view, you can edit the values in individual columns in the result set, add new rows, and delete existing rows.

Limitations in the Results Pane

- Results returned by table-valued functions can only be updated in some cases.
- Queries or views that include columns from more than one table or view cannot be updated.
- Results returned by a stored procedure cannot be updated.
- Queries or views using the GROUP BY or DISTINCT clauses are not updatable.

Navigating in the Results Pane

You can quickly navigate through the records using the navigation bar at the bottom of the Results pane.

There are buttons for going to the first and last records, the next and previous records, and for going to a particular record.

To go to a particular record, type the number of the row in the text box in the navigation bar and then press ENTER.

For information about using keyboard shortcuts in the Query and View Designer see Navigate in the Query and View Designer (Visual Database Tools).

Keeping the Results Set Synchronized with the Query Definition

While you are working on the results of a query or view it is possible for the records in the results pane to get out of synchronization with the query's definition. For example, if you run a query for four out of five columns in a table, and then use the Diagram pane to add the fifth column to the definition of the query, that fifth column's data will not automatically be added to the Results pane. To make the results pane reflect the new query definition, run the query again.

If a query changes, an alert icon and the text "Query Changed" appears in the lower-right corner of the results pane. The icon is repeated in the upper-left corner of the pane.

See Also

Create Queries (Visual Database Tools)
Run Queries (Visual Database Tools)
Design Queries and Views How-to Topics (Visual Database Tools)
Diagram Pane (Visual Database Tools)
Criteria Pane (Visual Database Tools)
Work with Data in the Results Pane (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Criteria Pane (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

The Criteria pane allows you to specify query options - such as which data columns to display, how to order the results, and what rows to select - by entering your choices into a spreadsheet-like grid. In the Criteria pane you can specify the following:

- Columns to display and column name aliases.
- The table that a column belongs to.
- Expressions for calculated columns.
- The sort order for the query.
- Search conditions.
- Grouping criteria, including aggregate functions to use for summary reports.
- New values for UPDATE or INSERT INTO queries.
- Target column names for INSERT FROM queries.

Changes you make in the Criteria pane are automatically reflected in the Diagram pane and SQL pane. Similarly, the Criteria pane is updated automatically to reflect changes made in the other panes.

About the Criteria Pane

The rows in the Criteria pane display the data columns used in your query; columns in the Criteria pane display query options.

The specific information that appears in the Criteria pane depends on the type of query you are creating.

If the Criteria pane is not visible, right-click the designer, point to **Pane**, and then click **Criteria**.

Options

Expand table

Column	Query type	Description
Column	All	Displays either the name of a data column used for the query or the expression for a computed column. This column is locked so that it is always visible as you scroll horizontally.
Alias	SELECT, INSERT FROM, UPDATE, MAKE TABLE	Specifies either an alternative name for a column or the name you can use for a computed column.
Table	SELECT, INSERT FROM, UPDATE, MAKE TABLE	Specifies the name of the table or table-structured object for the associated data column. This column is blank for computed columns.
Output	SELECT, INSERT FROM, MAKE TABLE	Specifies whether a data column appears in the query output. Note: If the database allows, you can use a data column for sort or search clauses without displaying it in the result set.
Sort Type	SELECT, INSERT FROM	Specifies that the associated data column is used to sort the query results and whether the sort is ascending or descending.
Sort Order	SELECT, INSERT FROM	Specifies the sort priority for data columns used to sort the result set. When you change the sort order for a data column, the sort order for all other columns is updated accordingly.
Group By	SELECT, INSERT FROM, MAKE TABLE	Specifies that the associated data column is being used to create an aggregate query. This grid column appears only if you have chosen **Group By** from the **Tools** menu or have added a GROUP BY clause to the SQL pane. By default, the value of this column is set to **Group By**, and the column becomes part of the GROUP BY clause. When you move to a cell in this column and select an aggregate function to apply to the associated data column, by default the resulting expression is added as an output column for the result set.
Criteria	All	Specifies a search condition (filter) for the associated data column. Enter an operator (the default is "=") and the value to search for. Enclose text values in single quotation marks. If the associated data column is part of a GROUP BY clause, the expression you enter is used for a HAVING clause. If you enter values for more than one cell in the **Criteria** grid column, the resulting search conditions are automatically linked with a logical AND.

Column	Query type	Description
		To specify multiple search condition expressions for a single database column, for example, (fname > 'A') AND (fname < 'M'), add the data column to the Criteria pane twice and enter separate values in the **Criteria** grid column for each instance of the data column.
Or...	All	Specifies an additional search condition expression for the data column, linked to previous expressions with a logical OR. You can add more **Or...** grid columns by pressing the TAB key in the rightmost **Or...** column.
Append	INSERT FROM	Specifies the name of the target data column for the associated data column. When you create an Insert From query, the Query and View Designer attempts to match the source to an appropriate target data column. If the Query and View Designer cannot choose a match, you must provide the column name.
New Value	UPDATE, INSERT INTO	Specifies the value to place into the associated column. Enter a literal value or an expression.

See Also

Design Queries and Views How-to Topics (Visual Database Tools)
Diagram Pane (Visual Database Tools)
Rules for Entering Search Values (Visual Database Tools)
Sort and Group Query Results (Visual Database Tools)
Results Pane (Visual Database Tools)
SQL Pane (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Display Query and View Designer Panes (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

The Query and View Designer is composed of four separate panes: the Diagram pane, the Criteria pane, the SQL pane, and the Results pane. Typically when you open a view or create a new query all four windows open, but if you need to open any of these windows, you can do so as long as the Query and View Designer is open.

To display Query and View Designer panes

1. Open a view or create a new query to open the Query and View Designer.
2. From the **Query Designer** menu, point to **Pane** and click the name of the pane you want to open.

See Also

Create Queries (Visual Database Tools)
Open the Query and View Designer (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Create Queries with Unnamed Parameters (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can create a query with an unnamed parameter by specifying a question mark (?) as a placeholder for a literal value. Query and View Designer will give it a temporary name. You can specify as many unnamed parameters in the query as you need.

When you run the query in the Query and View Designer, the Query Parameters Dialog Box is displayed with the temporary name.

To specify an unnamed parameter

1. Add the columns or expressions that you want to search to the Criteria pane. If you do not want the search columns or expressions to appear in the query output, remove them as output columns.

2. Locate the row containing the data column or expression to search, and then in the **Filter** grid column, enter a question mark (?).

   By default, the Query and View Designer adds the "=" operator. However, you can edit the cell to substitute ">", "<", or any other SQL comparison operator.

See Also

Query with Parameters (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Work with Diagram Layout (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

You can customize the way your database diagrams look.

In This Section

Customize the Amount of Information Displayed in Diagrams (Visual Database Tools)

Add Text Annotations to Diagrams (Visual Database Tools)

Autosize Selected Tables in Diagrams (Visual Database Tools)

Arrange Tables in Diagrams (Visual Database Tools)

Change the Font Color, Size, and Style in Diagrams (Visual Database Tools)

Reference

Related Sections

Work with Database Diagrams (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Add New Rows in the Results Pane (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can add new data either by typing it in or by pasting it from another program such as Notepad or Excel. A row to be pasted must have exactly the same number and types of columns as the table into which you are pasting. You can paste multiple rows into the Results pane at once.

For information about how to enter data see Rules for Updating Results (Visual Database Tools).

To add a new data row

1. Navigate to the bottom of the Results pane, where a blank row is available for adding a new data row.

   The initial values for all columns will be *NULL*.

   > **Tip**
   >
   > To go directly to the first empty row use the navigation bar at the bottom of the Results pane.

2. If you are pasting rows from the Clipboard, select the new row by clicking the button to its left.

   > **Note**
   >
   > If one or more of the rows you're pasting can't be committed to the database you will receive a message telling you which row(s) couldn't be committed.

3. Enter the data for the new row. If you are pasting, choose **Paste** from the **Edit** menu.

4. Leave that row to commit it to the database.

If an error occurs when you save the row the Query and View Designer displays a message and then returns you to the row you were editing. You can then:

- Resolve the error by making further edits in the row.
- Cancel the edit by pressing ESC. If you press ESC while in a cell that you changed, the changes for that cell are canceled. If you press ESC while in a cell you have not changed, the changes for the entire row are canceled.

See Also

Work with Data in the Results Pane (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Join Tables on Multiple Columns (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can join tables with multiple columns. That is, you can create a query that matches rows from the two tables only if they satisfy multiple conditions. If the database contains a relationship matching multiple foreign-key columns in one table to a multicolumn primary key in the other table, you can use this relationship to create a multicolumn join. For details, see Join Tables Automatically (Visual Database Tools).

Even if the database contains no multi-column foreign-key relationship, you can create the join manually.

To manually create a multicolumn join

1. Add to the Diagram pane the tables you want to join.
2. Drag the name of the first join column in the first table window and drop it onto the related column in the second table window. You cannot base a join on text, ntext, or image columns.

   > **Note**
   >
   > In general, the join columns must be of the same (or compatible) data types. For example, if the join column in the first table is a date, you must relate it to a date column in the second table. On the other hand, if the first join column is an integer, the related join column must also be of an integer data type, but it can be a different size. However, there may be cases where implicit data type conversions can join seemingly incompatible columns will work.
   >
   > The Query and View Designer will not check the data types of the columns you use to create a join, but when you execute the query, the database will display an error if the data types are not compatible.

3. Drag the name of the second join column in the first table window and drop it onto the related column in the second table window.
4. Repeat step 3 for each additional pair of join columns in the two tables.

5. Run the query.

See Also

Query with Joins (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Add Tables to Diagrams (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

You can add a table to your database diagram to edit its structure or relate it to other tables in your diagram. You can either add existing database tables to a diagram or insert a new table that has not yet been defined in the database.

To insert a new table into a diagram

1. Make sure you are connected to the database in which you want to create the table.

   To create a table in your current diagram, click the **New Table** button on the toolbar.

   -or-

   Right-click in the diagram and select **New Table**.

2. Modify or accept the system-assigned table name, in the **Choose Name** dialog box, and then choose **OK**.

   A new table appears in the diagram in Standard view.

3. In the first cell of the new table, type a column name. Then press the TAB key to move to the next cell.

4. Under **Data Type**, select a data type for the column. Each column must have a name and data type.

   You can set the column's other properties in Table Designer.

5. Repeat steps 3 and 4 for each column you want to add to the table.

> **Note**
>
> When you save your database diagram, the new table will be added to your database.

To add an existing table to a diagram

1. Make sure you are connected to the database whose tables you want to edit.
2. Select a table in the **Tables** folder.
3. Drag the table into your database diagram.
4. Release the mouse button.

> **Note**
>
> If relationships exist between the selected table and other tables in your diagram, relationship lines are automatically drawn.

To add related tables to a diagram

1. Select one or more tables with foreign key constraints in the database diagram.
2. Right-click any of the selected tables and choose **Add Related Tables**.

> **Note**
>
> Both those tables referenced by a foreign key constraint from the selected table(s) and those referencing the selected table(s) with a foreign key constraint are added to the diagram.

See Also

Work with Database Diagrams
Work with Tables in Database Diagram

Feedback

Was this page helpful? Yes No

Create Full-Text Search Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

Full-text searches use the CONTAINS predicate to locate rows that have specified text in a given column. Full-Text searches are only possible on columns that have active full-text indexes. If you attempt to use the CONTAINS clause on a column that does not have a currently active full-text index, you will receive an error. For more information on full-text indexes and the CONTAINS clause, see Full-Text Search (SQL Server) and CONTAINS (Transact-SQL).

To create a full-text search query

1. Open a query from Solution Explorer or create a new one.
2. In the WHERE clause of your query, use the CONTAINS function to search a full-text column.

See Also

Supported Query Types (Visual Database Tools)
Design Queries and Views How-to Topics (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Sort Multiple Columns in Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can specify the order in which you want to use the columns to sort the data by using the **Sort Order** column in the Criteria pane.

To sort query results by more than one column

1. In the **Criteria** pane, click the **Sort Order** field for the column upon which you want to sort.

2. Type an integer or select an integer from the drop-down list to indicate this column's place in the order of sorted columns.

   Repeat steps for each column by which you want to sort your results.

See Also

Sort and Group Query Results (Visual Database Tools)
Summarize Query Results (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Navigate in Database Diagram Designer (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

You can work in Database Diagram Designer using the keyboard as well as the mouse. Refer to the following tables for specific methods used in the Database Diagram Designer.

Expand table

To	Press
Access the Database Diagram menu	ALT+M
Move among tables and relationships if available	TAB, or SHIFT+TAB
Move between columns in a table	Arrow keys
Remove the selected table from the diagram	DELETE
Drop a table or relationship from the database	SHIFT+DELETE
Move tables across the diagram surface	CTRL+ UP ARROW or CTRL+DOWN ARROW
Scroll right/left and up/down	PAGE UP/DOWN for vertical, CTL+PAGE UP/DOWN for horizontal
Resize tables and annotations	SHIFT+ Arrow keys
Change the amount of information shown on a table	CTRL+T
Zoom	CTRL+ PLUS KEY or MINUS KEY

See Also

Design Database Diagrams (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Work with Data in the Results Pane (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

After you run a query or view, the results are shown in the Results pane. You can then work with those results. For example, you can add and delete rows, input or change data, and easily navigate through large results sets.

The following information can help you avoid problems and work effectively with your results sets.

Returning the results set

You can return results from either a query or a view and can choose whether to open just the results pane or all panes. In either case the query or view will open in Query and View Designer. The difference is that one opens with only the Results pane showing and the other opens with all windows that have been selected in the Options dialog box. The default is all four panes (Results, SQL, Diagram, and Criteria).

For more information see Open Queries (Visual Database Tools).

To change the design of the query or view so it returns a different set of results or returns records in a different order see the topics listed in Design Queries and Views How-to Topics (Visual Database Tools).

You can also determine whether to return all or part of the results set in two ways--stop the query as it runs or choose how much of results to return before the query is run.

Navigating in the results pane

You can quickly navigate through the records using the navigation bar at the bottom of the Results pane.

There are buttons for going to the first and last records, the next and previous records, and for going to a particular record.

To go to a particular record, type the number of the row in the text box in the navigation bar and then press ENTER.

For information about using keyboard shortcuts in the Query and View Designer see Navigate in the Query and View Designer (Visual Database Tools).

Committing changes to the database

The Results pane uses optimistic concurrency control so the grid shows a copy of the data in the database rather than an entirely live view. This way changes are only committed to the database after you move off of a row. This allows more than one user to work with the database at the same time. If there are conflicts (for example if another user changed the same row you changed and committed it to the database before you did) you will receive a message telling you of the conflict and offering resolutions.

Undo changes using ESC

You can only undo a change if it hasn't yet been committed to the database. The data is not committed if you haven't moved off of the record or if once you do move off the record you get an error message indicating the change won't be committed. If it hasn't been committed you can undo the change by using the ESC key.

To undo all changes in a row, move to a cell in that row that you have not edited and press the ESC key.

To undo changes to a particular cell that you have edited, move to that cell press the ESC key.

Adding or deleting data in the database

To see how your database design is working you may need to add sample data to the database. You can enter it into the results pane directly or you can copy it from another program, such as notepad or Excel, and paste it into the results pane.

In addition to copying rows into the Results pane you can add new records or modify or delete existing ones. For more information see Add New Rows in the Results Pane (Visual Database Tools), Delete Rows in the Results Pane (Visual Database Tools), and Edit Rows in the Results Pane (Visual Database Tools).

Tips for working with NULL values and empty cells

When you click on an empty row to add a new record, the initial value for all columns is *NULL*. If a column allows null values you can leave it as is.

If you want to replace a non-null value with null, type NULL in capital letters. The Results pane will give the word italic formatting to indicate that it is to be recognized as a null value rather than as a string.

To type in the string "null" type the letters without quotes. As long as at least one of the letters is in lower case, the value will be treated as a string rather than a null value.

Values for columns with a binary data type will have NULL values by default. These values can't be changed in the Results pane.

To input an empty space instead of using null, delete the existing text and move off of the cell.

Validating data

The Query and View Designer can validate some kinds of data against the columns properties. For example, if you enter "abc" into a column with a float data type, you will receive an error and the change will not be committed to the database.

The quickest way to see the data type of a column when you're in the Results pane is to open the Diagram pane and hover over the name of the column in the table or table-valued object.

> **Note**
>
> The maximum length the Results pane can show for a text data type is 2,147,483,647.

Keeping the results set synchronized with the query definition

While you're working on the results of a query or view it is possible for the records in the results pane to get out of synchronization with the queries definition. For example, if you ran a query for four out of five columns in a table, then used the Diagram pane to add the fifth column to the definition of the query, that fifth column's data will not automatically be added to the results pane. To make the results pane reflect the new query definition, run the query again.

You can tell if this happens--an alert icon and the text "Query Changed" appears in the lower-right corner of the results pane and the icon is repeated in the upper-left corner of the pane.

Reconciling changes made by multiple users

While you're working on the results of a query or view it is possible fore the records to be changed by a different user who is also working with the database.

If this happens you will receive a notice as soon as you move off of the cell with the conflict. You will then be able to override the other user's change, update your results pane with the other user's change, or keep editing your results pane without reconciling the differences. If you choose not to reconcile the differences your changes will not be committed to the database.

Limitations in the Results pane

What can not be updated

These tips may help you work successfully with data in the Results pane.

- Queries that include columns from more than one table or view can't be updated.
- Views can only be updated if the database constraints allow it.
- Results returned by a stored procedure can't be updated.
- Queries or views using the GROUP BY, DISTINCT, or TO XML clauses are not updatable.
- Results returned by table-valued functions can only be updated in some cases.
- Data in columns that result from an expression in the query.
- Data that was not successfully translated by the provider.

What can not be represented fully

What is returned to the Results pane from the database is greatly controlled by the provider for the data source you are using. The Results pane can't always translate the data from all database management systems. Here are come cases where this is so.

- Binary data types are often not useful for people working in the Results pane and they can take a very long time to download. So they are represented by *<Binary data>* or *Null*.
- Precision and scale can not always be preserved. For example, the Results pane supports a precision of 27. If data is of a data type with a greater precision, the data may be truncated or may be represented by *<Unable to read data>*.

See Also

Perform Basic Operations with Queries (Visual Database Tools)
Specify Search Criteria (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Clear Query Results (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

After running a query in the **Query and View Designer**, you can clear the Results pane. This procedure applies to queries created by using the **Open View** or **Open Table** options from Object Explorer. In SQL Server Management Studio, after running a query in the Query Editor, the Results pane cannot be cleared except by closing the Query Editor window or running another query.

> **Note**
>
> This will not undo the changes you've made to the database. For more information about how the Results pane works see Work with Data in the Results Pane (Visual Database Tools).

To clear query results of a view

- Right-click in the Results pane, point to **Pane**, and then click **Clear Results**.

  If a query is being executed when you clear the Results pane, the Query and View Designer stops the query.

See Also

Run Queries (Visual Database Tools)
Stop a Query (Visual Database Tools)
Work with Data in the Results Pane (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Use the Query and View Designer with International Data (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can use the Query and View Designer with data in any language and in any version of the Windows operating system. The following guidelines outline the differences you will notice and provide information about managing data in international applications.

Localized Information in the Criteria and SQL Panes

If you are using the Criteria pane to create your query, you can enter information in the format that corresponds to the Windows Regional Settings for you computer. For example, if you are searching for data, you can enter the data in the Criteria columns using whatever format you are accustomed to using, with these exceptions:

- Long data formats are not supported.
- Currency symbols should not be entered in the Criteria pane.
- Currency symbols will not display in the Results pane.

  > **Note**
  >
  > In the Results pane, you can actually enter the currency symbol that corresponds to the Windows Regional Settings for your computer, but the symbol will be removed and will not display in the Results pane.

- Unary minus always appears on the left side (for example, -1) regardless of the Regional Settings options.

In contrast, data and keywords in the SQL pane must always be in ANSI (U.S.) format. For example, as the Query and View Designer builds a query, it inserts the ANSI form of all SQL keywords such as SELECT and FROM. If you add elements to the statement in the SQL pane, be sure to use the ANSI standard form for the elements.

When you enter data using local-specific format in the Criteria pane, the Query and View Designer automatically translates it to ANSI format in the SQL pane. For example, if your Regional Settings are set to Standard German, you can enter data in the Criteria pane in a format such as "31.12.96." However, the date will appear in the SQL pane in ANSI datetime format as `{ ts '1996-12-31 00:00:00' }.` If you enter data directly in the SQL pane, you must enter it in ANSI format.

Sort Order

The sort order of data in your query is determined by the database. Options that you set in the Windows Regional Settings dialog box do not affect sort order for queries. Within any particular query, however, you can request that rows be returned in a particular order.

Using Double-Byte Characters

You can enter DBCS characters for literals and for database object names such as table and view names or aliases. You can also use DBCS characters for parameter names and parameter marker characters. However, you cannot use DBCS characters in SQL language elements such as function names or SQL keywords.

See Also

Design Queries and Views How-to Topics

Feedback

Was this page helpful? Yes No

Show Query Properties (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can display properties for a query in the Properties window from Query and View Designer.

To display properties for a query

- With a query open in Query and View Designer, from the **View** menu click **Properties Window**.

The properties for the query are now visible in the **Properties** window.

See Also

Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Understand Database Diagram Ownership (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

To use Database Diagram Designer it must first be set up by a member of the db_owner role (a role of SQL Server databases) to control access to diagrams. Each diagram has one and only one owner, the user who created it. For more information on setting up diagramming see Set Up Database Diagram Designer.

Diagram ownership considerations

Some points to keep in mind about diagram ownership:

- Although any user with access to a database can create a diagram, once the diagram has been created, the only users who can see it are the diagram creator and any member of the db_owner role. If the diagram creator is part of a user group and not a member of the db_owner role then they require their own login to view the diagram.
- Ownership of diagrams can only be transferred to members of the db_owner role. This is only possible if the previous owner of the diagram has been removed from the database.
- If the owner of a diagram has been removed from the database, the diagram will remain in the database until a member of the db_owner role attempts to open it. At that point the db_owner member can choose to take over ownership of the diagram.

See Also

Work with Database Diagrams
Set Up Database Diagram Designer

Feedback

Was this page helpful? Yes No

Create Table Aliases (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

Aliases can make it easier to work with table names. Using aliases is helpful when:

- You want to make the statement in the SQL Pane shorter and easier to read.
- You refer to the table name often in your query - such as in qualifying column names - and want to be sure you stay within a specific character-length limit for your query. (Some databases impose a maximum length for queries.)
- You are working with multiple instances of the same table (such as in a self-join) and need a way to refer to one instance or the other.

For example, you can create an alias `"e"` for a table name `employee_information`, and then refer to the table as `"e"` throughout the rest of the query.

To create an alias for a table or table-valued object

1. Add the table or table-valued object to your query.
2. In the **Diagram Pane**, right-click the object for which you want to create an alias, then select **Properties** from the shortcut menu.
3. In the **Properties** window, enter the alias in the **Alias** field.

See Also

Add Tables to Queries (Visual Database Tools)
Sort and Group Query Results (Visual Database Tools)
Summarize Query Results (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Parameter Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

In some cases you want to create a query that you can use many times, but with a different value each time. For example, you might frequently run a query to find all the `title_ids` written by one author. You could run the same query for each request, except that the author's ID or name would be different each time.

To create a query that can have different values at different times, you use parameters in the query. A parameter is a placeholder for a value that is supplied when the query runs. A SQL statement with a parameter might look like the following, where "?" represents the parameter for the author's ID:

```
SELECT title_id
FROM titleauthor
WHERE (au_id = ?)
```

Where You Can Use Parameters

You can use parameters as placeholders for literal values - for either text or numeric values. Most commonly, parameters are used as placeholders in search conditions for individual rows or for groups (that is, in the WHERE or HAVING clauses of a SQL statement).

You can use parameters as placeholders in expressions. For example, you might want to calculate discounted prices by supplying a different discount value each time you run a query. To do so, you could specify the following expression:

```
(price * ?)
```

Specifying Unnamed and Named Parameters

You can specify two types of parameters: unnamed and named. An unnamed parameter is a question mark (?) that you put anywhere in the query that you want to prompt for or substitute a literal value. For example, if you use an unnamed parameter to search for an author's id in the `titleauthor` table, the resulting statement in the SQL Pane might look like this:

```
SELECT title_id
FROM titleauthor
WHERE (au_id = ?)
```

When you run the query in the Query and View Designer, the Query Parameters dialog box appears with "?" as the name of the parameter.

Alternatively, you can assign a name to a parameter. Named parameters are particularly useful if you have multiple parameters in a query. For example, if you use named parameters to search for an author's first and last names in the `authors` table, the resulting statement in the SQL pane might look like this:

```
SELECT au_id
FROM authors
WHERE au_fname = %first name% AND
      au_lname = %last name%
```

> **Tip**
>
> You must define prefix and suffix characters before creating a named parameter query.

When you run the query in the Query and View Designer, the Query Parameters dialog box appears with a list of named parameters.

See Also

Query with Parameters (Visual Database Tools)
Supported Query Types (Visual Database Tools)
Design Queries and Views How-to Topics (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Copy Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can copy the body of a query into a new query to use as a starting point.

To copy a query

1. From the **View** menu, click Solution Explorer.
2. In Solution Explorer, right-click the query you want to copy and from the shortcut menu choose **Open**.
3. In the SQL pane, copy the portion of the query you want to copy.
4. Create a new query or open the query where you want to paste the copied SQL.
5. Position the cursor where you want to add the SQL code.
6. Right-click and from the shortcut menu click **Paste**.

See Also

Rename Queries (Visual Database Tools)
Clear Query Results (Visual Database Tools)
Save Queries (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Sort with ORDER BY (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can sort query results by one or more of the columns in the returned rows by using an ORDER BY clause. You can define an ORDER BY clause by choosing options in the Criteria Details pane.

To sort a query using an ORDER BY clause

1. Open a query or create a new one.
2. In the Criteria Pane, click the **Sort Type** column for the row corresponding to the column that you want to use to sort your query results.
3. Choose *Ascending* or *Descending* from the drop-down list.

> **Note**
>
> Clearing the **Sort Type** entry for a column removes that column from the ORDER BY clause.

Notice that as you work in the Criteria pane, your query's UNION clause changes to match your most recent actions.

> **Note**
>
> When sorting results by more than one column, specify the order in which columns are searched relative to each other by using the **Sort Order** column. For more information, see **How to: Sort Multiple Columns in Queries**.

See Also

Sort and Group Query Results (Visual Database Tools)
Summarize Query Results (Visual Database Tools)
Design Queries and Views How-to Topics (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Combine Conditions When AND Has Precedence (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

To combine conditions with AND, you add the column to the query twice--once for each condition. To combine conditions with OR, you put the first one in the Filter column and additional conditions into an **Or...** column.

For example, imagine that you want to find either employees who have been with the company for more than five years in lower-level jobs or employees with middle-level jobs regardless of their hire date. This query requires three conditions, two of them linked with AND:

- Employees with a hire date earlier than five years ago AND with a job level of 100.

  -or-

- Employees with a job level of 200.

To combine conditions when AND has precedence

1. In the Criteria pane, add the data columns you want to search. If you want to search the same column using two or more conditions linked with AND, you must add the data column name to the grid once for each value you want to search.

2. In the **Filter** column, enter all the conditions that you want to link with AND. For example, to link conditions with AND that search the `hire_date` and `job_lvl` columns, enter the values `< '1/1/91'` and `= 100`, respectively, in the Filter column.

   These grid entries produce the following WHERE clause in the statement in the SQL Pane:

   ```
   WHERE (hire_date < '01/01/91') AND
     (job_lvl = 100)
   ```

3. In the **Or...** grid column, enter conditions that you want to link with OR. For example, to add a condition that searches for another value in the `job_lvl` column, enter an additional value in the **Or...** column, such as `= 200`.

   Adding a value in the **Or...** column adds another condition to the WHERE clause in the statement in the SQL pane:

```
WHERE (hire_date < '01/01/91' ) AND
  (job_lvl = 100) OR
  (job_lvl = 200)
```

See Also

Combine Conditions When OR Has Precedence
Conventions for Combining Search Conditions in the Criteria Pane
Rules for Entering Search Values
Specify Search Criteria

Feedback

Was this page helpful? Yes No

Create Make Table Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can copy rows into a new table using a Make Table query, which is useful for creating subsets of data to work with or copying the contents of a table from one database to another. A Make Table query is similar to an Insert Results query but creates a new table to copy rows into.

When you create a Make Table query, you specify:

- The name of the new database table (the destination table).
- The table or tables to copy rows from (the source table). You can copy from a single table or from joined tables.
- The columns in the source table whose contents you want to copy.
- Sort order, if you want to copy the rows in a particular order.
- Search conditions to define the rows you want to copy.
- Group By options, if you want to copy only summary information.

For example, the following query creates a new table called `uk_customers` and copies information from the `customers` table to it:

```
SELECT *
INTO uk_customers
FROM customers
WHERE country = 'UK'
```

In order to use a Make Table query successfully:

- Your database must support the SELECT...INTO syntax.
- You must have permission to create a table in the target database.

To create a Make Table query

1. Add the source table or tables to the Diagram pane.
2. From the **Query Designer** menu, point to **Change Type**, and then click **Make Table**.
3. In the **Make Table** dialog box, type the name of the destination table. The Query and View Designer does not check whether the name is already in use or whether you have permission to create the table.

   To create a destination table in another database, specify a fully qualified table name including the name of the target database, the owner (if required), and the name of the table.
4. Specify the columns to copy by adding them to the query. For details, see Add Columns to Queries. Columns will be copied only if you add them to the query. To copy entire rows, choose * **(All Columns)**.

   The Query and View Designer adds the columns you choose to the **Column** column of the Criteria pane.
5. If you want to copy rows in a particular order, specify a sort order. For details, see **Sorting and Grouping Query Results**.
6. Specify the rows to copy by entering search conditions. For details, see Specify Search Criteria.

   If you do not specify a search condition, all rows from the source table will be copied to the destination table.

   > **Note**
   >
   > When you add a column to search to the Criteria pane, the Query and View Designer also adds it to the list of columns to copy. If you want to use a column for searching but not copy it, clear the check box next to the column name in the rectangle representing the table or table-structured object.
7. If you want to copy summary information, specify Group By options. For details, see Summarize Query Results.

When you execute a Make Table query, no results are reported in the Results Pane. Instead, a message appears indicating how many rows were copied.

See Also

Design Queries and Views How-to Topics

[Types of Queries(../../ssms/visual-db-tools/types-of-queries-visual-database-tools.md)

Feedback

Was this page helpful? Yes No

Rules for Entering Search Values (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

This topic discusses the conventions you must use when entering the following types of literal values for a search condition:

- Text values
- Numeric values
- Dates
- Logical values

> **Note**
>
> The information in this topic is derived from the rules for standard SQL-92. However, each database can implement SQL in its own way. Therefore, the guidelines provided here might not apply in every case. If you have questions about how to enter search values for a particular database, refer to the documentation for the database that you are using.

Searching on Text Values

The following guidelines apply when you enter text values in search conditions:

- **Quotation marks** Enclose text values in single quotation marks, as in this example for a last name:

  ```
  'Smith'
  ```

  If you are entering a search condition in the Criteria Pane, you can simply type the text value and the Query and View Designer will automatically put single quotation marks around it.

> ⓘ **Note**
>
> In some databases, terms in single quotation marks are interpreted as literal values, whereas terms in double quotation marks are interpreted as database objects such as column or table references. Therefore, even though the Query and View Designer can accept terms in double quotation marks, it might interpret them differently than you expect.

- **Embedding apostrophes** If the data you are searching for contains a single quotation mark (an apostrophe), you can enter two single quotation marks to indicate that you mean the single quotation mark as a literal value and not a delimiter. For example, the following condition searches for the value "Swann's Way:"

  ```
  ='Swann''s Way'
  ```

- **Length limits** Do not exceed the maximum length of the SQL statement for your database when entering long strings.

- **Case sensitivity** Follow the case sensitivity rules for the database you are using. The database you are using determines whether text searches are case sensitive. For example, some databases interpret the operator "=" to mean an exact case-sensitive match, but others will allow matches on any combination of uppercase and lowercase characters.

  If you are unsure about whether the database uses a case-sensitive search, you can use the UPPER or LOWER functions in the search condition to convert the case of the search data, as illustrated in the following example:

  ```
  WHERE UPPER(lname) = 'SMITH'
  ```

Searching on Numeric Values

The following guidelines apply when you enter numeric values in search conditions:

- **Quotation marks** Do not enclose numbers in quotation marks.

- **Non-numeric characters** Do not include non-numeric characters except the decimal separator (as defined in the **Regional Settings** dialog box of Windows Control Panel) and negative sign (-). Do not include digit grouping symbols (such as a comma between thousands) or currency symbols.
- **Decimal marks** If you are entering whole numbers, you can include a decimal mark, whether the value you are searching for is an integer or a real number.
- **Scientific notation** You can enter very large or very small numbers using scientific notation, as in this example:

```
> 1.23456e-9
```

Searching on Dates

The format you use to enter dates depends on the database you are using and in what pane of the Query and View Designer you are entering the date.

> ⓘ **Note**
>
> If you don't know which format your data source uses, type a date into the filter column of the Criteria pane in any format familiar to you. The designer will convert most of such entries into the appropriate format.

The Query and View Designer can work with the following date formats:

- **Locale-specific** The format specified for dates in the **Windows Regional Settings Properties** dialog box.
- **Database-specific** Any format understood by the database.
- **ANSI standard date** A format that uses braces, the marker 'd' to designate the date, and a date string, as in the following example:

```
{ d '1990-12-31' }
```

- **ANSI standard datetime** Similar to ANSI-standard date, but uses 'ts' instead of 'd' and adds hours, minutes, and seconds to the date (using a 24-hour clock), as in

this example for December 31, 1990:

```
{ ts '1990-12-31 00:00:00' }
```

In general, the ANSI standard date format is used with databases that represent dates using a true date data type. In contrast, the datetime format is used with databases that support a datetime data type.

The following table summarizes the date format that you can use in different panes of the Query and View Designer.

Expand table

Pane	Date format
Criteria	Locale-specific Database-specific ANSI standard Dates entered in the Criteria Pane are converted to a database-compatible format in the SQL pane.
SQL	Database-specific ANSI standard
Results	Locale-specific

Searching on Logical Values

The format of logical data varies from database to database. Very frequently, a value of False is stored as zero (0). A value of True is most frequently stored as 1 and occasionally as -1. The following guidelines apply when you enter logical values in search conditions:

- To search for a value of False, use a zero, as in the following example:

```
SELECT * FROM authors
WHERE contract = 0
```

- If you are not sure what format to use when searching for a True value, try using 1, as in the following example:

```
SELECT * FROM authors
WHERE contract = 1
```

- Alternatively, you can broaden the scope of the search by searching for any non-zero value, as in the following example:

```
SELECT * FROM authors
WHERE contract <> 0
```

See Also

Specify Search Criteria (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Diagram Pane (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

The Diagram pane presents a graphic display of the tables or table-valued objects you have selected from the data connection. It also shows any join relationships among them.

In the Diagram pane you can:

- Add or remove tables and table-valued objects and specify data columns for output.
- Create or modify joins between tables and table-valued objects.

When you make a change in the Diagram pane, the Criteria pane and SQL pane are updated to reflect your change. For example, if you select a column for output in a table or table-valued object window in the Diagram pane, the Query and View Designer adds the data column to the Criteria pane and to the SQL statement in the SQL pane.

Each table or table-valued object appears as a separate window in the Diagram pane. The icon in the title bar of each rectangle indicates what type of object the rectangle represents, as illustrated in the following table.

Options

Tables
Lists the tables you can add to the Diagram pane. To add a table, select it and click **Add**. To add several tables at once, select them and click **Add**.

Views
Lists the views you can add to the Diagram pane. To add a view, select it and click **Add**. To add several views at once, select them and click **Add**.

Functions
Lists the user-defined functions you can add to the Diagram pane. To add a function, select it and click **Add**. To add several functions at once, select them and click **Add**.

Local Tables
Lists tables created by queries rather than the ones that belong to the database.

Synonyms

Lists the synonyms you can add to the Diagram pane. To add a synonym, select it and click **Add**. To add several synonyms at once, select them and click **Add**.

Expand table

Icon	Object type
	Table
	Query or View
	Linked Table
	User-Defined Function
	Linked View

Each rectangle shows the data columns for the table or table-valued object. Check boxes and symbols appear next to the names of columns to indicate how the columns are used in the query. ToolTips display information such as data type and size for columns.

The following table lists the check boxes and symbols used in the rectangle for each table or table-valued object.

Expand table

Check box or symbol	Description
     	Specifies whether a data column appears in the query result set (Select query) or is used in an Update, Insert From, Make Table, or Insert Into query. Select the column to add it to the results. If **(All Columns)** is selected, all data columns appear in the output. The icon used with the check box changes according to the type of query you are creating. When creating a Delete query, you cannot select individual columns.
 	Indicates that the data column is being used to order the query results (is part of an ORDER BY clause). The icon appears as A-Z if the sort order is ascending or Z-A if sort order is descending.
	Indicates that the data column is being used to create a grouped result set (is part of a GROUP BY clause) in an aggregate query.
	Indicates that the data column is included in a search condition for the query (is part of a WHERE or HAVING clause).

Check box or symbol	Description
Σ	Indicates that the contents of the data column are being summarized for output (are included in a SUM, AVG, or other aggregate function).

> **Note**
>
> Query and View Designer will not display data columns for a table or table-valued object if you do not have sufficient access rights to it or if the database driver cannot return information about it. In such cases, the Query and View Designer displays only a title bar for the table or table-structured object.

Joined Tables on the Diagram Pane

If the query involves a join, a join line appears between the data columns involved in the join. If the joined data columns are not displayed (for example, the table or table-valued object window is minimized or the join involves an expression), the Query and View Designer places the join line in the title bar of the rectangle representing the table or table-valued object. The Query and View Designer displays one join line for each join condition.

The shape of the icon in the middle of the join line indicates how the tables or table-structured objects are joined. If the join clause uses an operator other than equal (=), the operator is displayed in the join line icon. The following table lists the icons that can be displayed in a join line.

Expand table

Join line icon	Description
	Inner join (created using equal sign).
	Inner join based on the "greater than" operator. (The operator displayed in the join line icon reflects the operator used in the join.)
	Outer join in which all rows from the table represented on the left will be included, even if they do not have matches in the related table.
	Outer join in which all rows from the table represented on the right will be included, even if they do not have matches in the related table.

Join line icon	Description
	A full outer join in which all rows from both tables will be included, even if they do not have matches in the related table.

Icons on the ends of the join line indicate the type of join. The following table lists the types of joins and the icons that can be displayed on the ends of the join line.

Expand table

Icon on ends of join line	Description
	One-to-one join
	One-to-many join
	Query and View Designer cannot determine join type

See Also

Design Queries and Views How-to Topics (Visual Database Tools)
Criteria Pane (Visual Database Tools)
Sort and Group Query Results (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Create Subqueries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can use the results of one query as the input for another. You can use the results of a subquery as a statement that uses the IN( ) function, the EXISTS operator, or the FROM clause.

You can create a subquery by entering it directly into the SQL pane or by copying a query and pasting it into another.

To define a subquery in the SQL pane

1. Create the primary query.
2. In the SQL pane, select the SQL statement, and then use **Copy** to move the query to the Clipboard.
3. Start the new query, and then use **Paste** to move the first query into the new query's WHERE or FROM clause.

   For example, imagine you have two tables, `products` and `suppliers`, and you want to create a query showing all products for suppliers in Sweden. Create the first query on the `suppliers` table to find all Swedish suppliers:

   ```
   SELECT supplier_id
   FROM supplier
   WHERE (country = 'Sweden')
   ```

   Use the Copy command to move this query to the Clipboard. Create the second query using the `products` table, listing the information you need about products:

   ```
   SELECT product_id, supplier_id, product_name
   FROM products
   ```

In the SQL pane, add a WHERE clause to the second query, then paste the first query from the Clipboard. Place parentheses around the first query, so that the end result looks like this:

```
SELECT product_id, supplier_id, product_name
FROM products
WHERE supplier_id IN
   (SELECT supplier_id
  FROM supplier
  WHERE (country = 'Sweden'))
```

See Also

Supported Query Types (Visual Database Tools)
Specify Search Criteria (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Summarize or Aggregate Values Using Custom Expressions (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

In addition to using aggregate functions to aggregate data, you can create custom expressions to produce aggregate values. You can use custom expressions in place of aggregate functions anywhere in an aggregate query.

For example, in the `titles` table you might want to create a query that shows not just the average price, but what the average price would be if it were discounted.

You cannot include an expression that is based on calculations involving only individual rows in the table; the expression must be based on an aggregate value, because only the aggregate values are available at the time the expression is calculated.

To specify a custom expression for a summary value

1. Specify the groups for your query. For details, see Group Rows in Query Results (Visual Database Tools).
2. Move to a blank row of the Criteria pane, and then type the expression in the **Columns** column.

   The Query and View Designer automatically assigns a column alias to the expression to create a useful column heading in query output. For more details, see Create Column Aliases (Visual Database Tools).
3. In the **Group By** column for the expression, select **Expression**.
4. Run the query.

See Also

Sort and Group Query Results (Visual Database Tools)
Summarize Query Results (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Multiuser Environments (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

A multiuser environment is one in which other users can connect and make changes to the same database that you are working with. As a result, several users might be working with the same database objects at the same time. Thus, a multiuser environment introduces the possibility of the database being affected by changes made by other users while you are making changes, and vice versa.

A key issue when working with databases in a multiuser environment is access permissions. The permissions you have for the database determine the extent of the work you can do with the database. For example, to make changes to objects in a database, you must have the appropriate write permissions for the database. For more information about permissions in your database, see your database documentation. For more information see Permissions and Visual Database Tools (Visual Database Tools).

When you save changes made to tables, Table Designer verifies that the database has not been modified since you last saved changes. If another user has made changes, you will be notified that the database has been modified. You may need to reconcile these changes. For more information, see Reconcile Changes Made by Multiple Users (Visual Database Tools).

In a multiuser environment there are special considerations to keep in mind to avoid conflicting changes. For more information, see Visual Database Tools.

One way to avoid problems is to work in a copy of the database, such as a test database, when you make changes, then you can create a change script that you can run to make those changes on the original database after resolving conflicts offline. For more information see Visual Database Tools.

Feedback

Was this page helpful? Yes No

Customize the Amount of Information Displayed in Diagrams (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

When you add a table to a database diagram it appears with a set number of properties showing. You can choose from a menu of variations on how much information shows or you can create a custom set of properties.

The choices are:

- Standard
- Column Names
- Keys
- Name Only
- Custom
- Modify Custom

To choose a table view

1. Right-click a table in the Database Diagram Designer.
2. From the shortcut menu, point to Table View and then click the view you want.

To create a custom table view

1. Right-click a table in Database Diagram Designer.
2. From the shortcut menu, point to **Table View** and then click **Modify Custom**.
3. In the **Column Selection** dialog box, choose which properties you want to show when **Custom** in chosen from the **Table View** menu.

See Also

Work with Database Diagrams (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Delete a Database Diagram (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

When you no longer need a database diagram, you can delete it.

To delete a database diagram

1. In Object Explorer, expand the **Database Diagrams** folder.
2. Right-click the database diagram that you want to delete.
3. Choose **Delete** on the shortcut menu.
4. A message prompts you to confirm the deletion. Choose **Yes**.

The database diagram is deleted from the database. When you delete a database diagram, the tables in the diagram are not deleted.

See Also

Work with Database Diagrams (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Query with Parameters (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

In many instances, it is useful to create a query that you can reuse many times in your applications with different input each time. In this section you will find information about how to create parameter queries for these general purposes.

In This Section

Parameter Queries (Visual Database Tools)
Learn about general-purpose queries.

Create Queries with Unnamed Parameters (Visual Database Tools)
Create a query using unnamed parameters.

Related Sections

Perform Basic Operations with Queries (Visual Database Tools)
Provides links to topics covering the most common query tasks.

Specify Search Criteria (Visual Database Tools)
Provides links to topics covering the various ways to search and how to search for what you want.

Feedback

Was this page helpful? Yes No

Work with database diagrams (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

You can create a visual diagram of your database using the Database Designer. The following articles cover the basics of using the designer to work with database diagrams.

In this section

- Create a New Database Diagram (Visual Database Tools)
- Save Database Diagrams (Visual Database Tools)
- Open Database Diagrams (Visual Database Tools)
- Delete a Database Diagram (Visual Database Tools)
- Reconcile a Database Diagram with a Modified Database (Visual Database Tools)
- Show Diagram Properties (Visual Database Tools)

Related content

- Work with Tables in Database Diagram (Visual Database Tools)
- Work with Diagram Layout (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Create Insert Values Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can create a new row in the current table using an Insert Values query. When you create an Insert Values query, you specify:

- The database table to add the row to.
- The columns whose contents you want to add.
- The value or expression to insert into the individual columns.

For example, the following query adds a row to the `titles` table, specifying values for the title, type, publisher, and price:

```
INSERT INTO titles
         (title_id, title, type, pub_id, price)
VALUES   ('BU9876', 'Creating Web Pages', 'business', '1389', '29.99')
```

When you create an Insert Values query, the Criteria pane changes to reflect the only options available for inserting a new row: the column name and the value to insert.

> **⊗ Caution**
>
> You cannot undo the action of executing an Insert Values query. As a precaution, back up your data before executing the query.

To create an Insert Values query

1. Add the table you want to update to the Diagram pane.
2. From the **Query Designer** menu point to **Change Type**, and then click **Insert Values**.

   > **ⓘ Note**

If more than one table is displayed in the Diagram pane when you start the Insert Values query, the Query and View Designer displays the Choose Target Table for Insert Values dialog box to prompt you for the name of the table to update.

3. In the Diagram pane, click the check box for each column for which you want to supply new values. Those columns will show in the Criteria pane. Columns will be updated only if you add them to the query.

4. In the **New Value** column of the Criteria pane, enter the new value for the column. You can enter literal values, column names, or expressions. The value must match (or be compatible with) the data type of the column you are updating.

   ⊗ Caution

   The Query and View Designer cannot check that a value fits within the length of the column you are inserting. If you provide a value that is too long, it might be truncated without warning. For example, if a `name` column is 20 characters long but you specify an insert value of 25 characters, the last 5 characters might be truncated.

When you execute an Insert Values query, no results are reported in the Results Pane. Instead, a message appears indicating how many rows were changed.

See Also

Supported Query Types (Visual Database Tools)
Design Queries and Views How-to Topics (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Specify Multiple Search Conditions for One Column (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

In some instances, you might want to apply a number of search conditions to the same data column. For example, you might want to:

- Search for several different names in an `employee` table or for employees who are in different salary ranges. This type of search requires an OR condition.
- Search for a book title that both starts with the word "The" and contains the word "Cook." This type of search requires an AND condition.

> **Note**
>
> The information in this topic applies to search conditions in both the WHERE and HAVING clauses of a query. The examples focus on creating WHERE clauses, but the principles apply to both types of search conditions.

To search for alternative values in the same data column, you specify an OR condition. To search for values that meet several conditions, you specify an AND condition.

Specifying an OR Condition

Using an OR condition enables you to specify several alternative values to search for in a column. This option expands the scope of the search and can return more rows than searching for a single value.

> **Tip**
>
> You can often use the IN operator instead to search for multiple values in the same data column.

To specify an OR condition

1. In the Criteria Pane, add the column to search.

2. In the **Filter** column for the data column you just added, specify the first condition.

3. In the **Or...** column for the same data column, specify the second condition.

The Query and View Designer creates a WHERE clause that contains an OR condition such as the following:

```
SELECT fname, lname
FROM employees
WHERE (salary < 30000) OR (salary > 100000)
```

Specifying an AND Condition

Using an AND condition enables you to specify that values in a column must meet two (or more) conditions for the row to be included in the result set. This option narrows the scope of the search and usually returns fewer rows than searching for a single value.

> **Tip**
>
> If you are searching for a range of values, you can use the BETWEEN operator instead of linking two conditions with AND.

To specify an AND condition

1. In the Criteria pane, add the column to search.

2. In the **Filter** column for the data column you just added, specify the first condition.

3. Add the same data column to the Criteria pane again, placing it in an empty row of the grid.

4. In the **Filter** column for the second instance of the data column, specify the second condition.

The Query Designer creates a WHERE clause that contains an AND condition such as the following:

```
SELECT title_id, title
FROM titles
```

```
WHERE (title LIKE '%Cook%') AND
  (title LIKE '%Recipe%')
```

See Also

Conventions for Combining Search Conditions in the Criteria Pane (Visual Database Tools)
Specify Search Criteria (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Map Many-to-Many Relationships (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

Many-to-many relationships let you relate each row in one table to many rows in another table, and vice versa. For example, you could create a many-to-many relationship between the `authors` table and the `titles` table to match each author to all of their books and to match each book to all of its authors. Creating a one-to-many relationship from either table would incorrectly indicate that every book can have only one author, or that every author can write only one book.

Many-to-many relationships between tables are accommodated in databases by means of junction tables. A junction table contains the primary key columns of the two tables you want to relate. You then create a relationship from the primary key columns of each of those two tables to the matching columns in the junction table. In the pubs database, the `titleauthor` table is a junction table.

To create a many-to-many relationship between tables

1. In your database diagram, add the tables that you want to create a many-to-many relationship between.
2. Create a third table by right-clicking the diagram and choosing **New Table** from the shortcut menu. This will become the junction table.
3. In the **Choose Name** dialog box, change the system-assigned table name. For example, the junction table between the `titles` table and the `authors` table is now named `titleauthors`.
4. Copy the primary key columns from each of the other two tables to the junction table. You can add other columns to this table, just as you can to any other table.
5. In the junction table, set the primary key to include all the primary key columns from the other two tables. For details, see How to: Create Primary Keys.
6. Define a one-to-many relationship between each of the two primary tables and the junction table. The junction table should be at the "many" side of both of the

relationships you create. For details, see How to: Create Relationships Between Tables.

> **Note**
>
> The creation of a junction table in a database diagram does not insert data from the related tables into the junction table. For information about inserting data into a table, see Create Insert Results Queries (Visual Database Tools).

See Also

Work with Database Diagrams (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Create Relationships Between Tables on a Diagram (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can create relationships between columns in different tables in the Diagram Designer by dragging columns between tables.

To create a relationship graphically

1. In Database Designer, click the row selector for one or more database columns that you want to relate to a column in another table.
2. Drag the selected column(s) to the related table.
3. Two dialog boxes appear: **Foreign Key Relationship** and **Tables and Columns**, with the latter appearing in the foreground.
4. **Relationship name** has a system-provided name in the format FK_*localtable_foreigntable*. You may change this value.
5. Verify that **Primary key table** specifies the correct table.
6. The grid lists the local columns and their matching foreign columns. You can add or remove table columns or change mappings.
7. Choose **OK**.

   The **Foreign Key Relationship** dialog box appears. **Selected Relationship** shows the relationship you have created.
8. Change properties for the relationship in the grid.
9. Choose **OK** to create the relationship.

   Database Designer shows a relationship between the columns you chose.

See Also

Working with Constraints
Work with Tables in Database Diagram (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Specify Conditions for Groups (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can limit the groups that appear in a query by specifying a condition that applies to groups as a whole - a HAVING clause. After the data has been grouped and aggregated, the conditions in the HAVING clause are applied. Only the groups that meet the conditions appear in the query.

For example, you might want to see the average price of all books for each publisher in a `titles` table, but only if the average price exceeds $10.00. In that case, you could specify a HAVING clause with a condition such as `AVG(price) > 10`.

> **Note**
>
> In some instances, you might want to exclude individual rows from groups before applying a condition to groups as a whole. For details, see Use HAVING and WHERE Clauses in the Same Query (Visual Database Tools).

You can create complex conditions for a HAVING clause by using AND and OR to link conditions. For details about using AND and OR in search conditions, see Specify Multiple Search Conditions for One Column (Visual Database Tools).

To specify a condition for a group

1. Specify the groups for your query. For details, see Group Rows in Query Results (Visual Database Tools).

2. If it is not already in the Criteria pane, add the column on which you want to base the condition. (Most often the condition involves a column that is already a group or summary column.) You cannot use a column that is not part of an aggregate function or of the GROUP BY clause.

3. In the **Filter** column, specify the condition to apply to the group.

   The Query and View Designer automatically creates a HAVING clause in the statement in the SQL pane, such as in the following example:

   ```
   ```

```
SELECT pub_id, AVG(price)
FROM titles
GROUP BY pub_id
HAVING (AVG(price) > 10)
```

4. Repeat steps 2 and 3 for each additional condition you want to specify.

See Also

Sort and Group Query Results (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Specify Search Criteria (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can use search criteria to restrict the number of rows returned by a query.

For details about the particular steps to creating search criteria, refer to the topics listed in the following table.

In This Section

Rules for Entering Search Values (Visual Database Tools)
Explains how to enter text, numbers, dates, or logical values.

Conventions for Combining Search Conditions in the Criteria Pane (Visual Database Tools)
Explains the concepts behind using the AND and OR operators.

Specify Search Conditions (Visual Database Tools)
Explains the basics of choosing search criteria to get just the data you want.

Specify Multiple Search Conditions for One Column (Visual Database Tools)
Explains how to create multiple search conditions to the same data column.

Specify Multiple Search Conditions for Multiple Columns (Visual Database Tools)
Explains how to include several data columns as part of the search condition for a query.

Specify the TOP Clause in Queries (Visual Database Tools)
Explains how to have only a given number or percentage of rows returned.

Use HAVING and WHERE Clauses in the Same Query (Visual Database Tools)
Explains how and why to use both of these clauses in a query.

Select Rows That Do Not Match a Value (Visual Database Tools)
Explains how to return all rows where the value of a give column does not match the value you provide in the query statement.

Include or Exclude Rows (Visual Database Tools)
Explains the concepts behind clauses and operators used in queries.

Exclude Duplicate Rows (Visual Database Tools)
Explains how to filter duplicate rows out of Select queries.

Combine Conditions When AND Has Precedence (Visual Database Tools)
Explains why and how to use the AND operator to filter query results.

Combine Conditions When OR Has Precedence (Visual Database Tools)
Explains why and how to use the OR operator to filter query results.

Create Subqueries (Visual Database Tools)
Explains how to create subqueries, or nested queries.

Related Sections

Perform Basic Operations with Queries (Visual Database Tools)
Provides links to topics with steps for the most common querying tasks.

Types of Queries (Visual Database Tools)
Provides links to topics explaining the supported query types.

Sort and Group Query Results (Visual Database Tools)
Provides links to topics with steps for sorting and grouping the results of a query.

Summarize Query Results (Visual Database Tools)
Provides links to topics with steps for summarizing results, including NULL and Nonnumeric columns.

Design Queries and Views How-to Topics (Visual Database Tools)
Provides links to topics and sections covering work you can do in queries and views using the Query and View Designer.

Feedback

Was this page helpful? Yes No

Arrange Tables in Diagrams (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

You can quickly arrange the tables in a database diagram to see each table and relationship clearly.

To arrange tables in diagrams

1. Right-click in the empty space in the Database Diagram window.
2. From the shortcut menu click **Arrange Tables**.

See Also

Work with Database Diagrams (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Walkthrough: Adding and Changing a Database Diagram

Article • 02/13/2025

Applies to: SQL Server

This walkthrough illustrates how to create and modify a database diagram and make changes to the database through the database diagrams component. You will see how to add tables to diagrams, create relationships between tables, create constraints and indexes on columns, and change the level of information you see for each table.

Prerequisites

In order to complete this walkthrough, you will need:

- Access to SQL Server with the `AdventureWorks2022` sample database
- An account with database owner **dbo** privileges

> **Note**
>
> If you attempt to make changes when using an account without sufficient privileges to make changes to tables, then an error message appears.

Creating a Diagram

To create a new database diagram

1. On the **View** menu, click **Object Explorer**.
2. Open the Databases node and then open the `AdventureWorks2022` node.
3. Right-click the Database Diagrams node and choose **New Database Diagram**.

   If the database does not have objects necessary to create diagrams, the following message appears: **This database does not have one or more of the support objects required to use database diagramming. Do you wish to create them?** Choose **Yes**.

   The **Add Table** dialog box appears.

4. Select **AddressType (Person)** and **Address (Person)** and click **Add**.

   Two tables are added to the diagram.

5. Close the **Add Table** dialog box.

To view different column data

1. Right-click the `Address` table. On the shortcut menu, point to **Table View**, and then click **Standard**.

   The table grid shows three columns: **Column Name**, **Data Type**, and **Allow Nulls**.

2. Right-click the `Address` table, click **Table View** and select **Keys**.

   The table grid shows one column, with the table-column names. Only those columns participating in indexes appear.

Creating New Tables

To create tables within Diagram Designer

1. Right-click the Diagram Designer outside the existing tables and choose **New Table**.

2. In the **Choose Name** dialog box, click **OK** to accept the default name **Table1**.

   A new table grid appears with three columns: **Column Name**, **Data Type**, and **Allow Nulls**.

3. Add the following information to **Table1**:

Expand table

Column Name	Data Type	Allow Nulls
T1col1	**int**	checked
T1col2	**varchar(50)**	checked
T1col3	**float**	checked

4. Right-click `T1col1` and select **Set Primary Key**.

   A key icon will appear beside the column name.

5. From the **File** menu, click **Save Diagram1**.
6. In the **Choose Name** dialog box, click **OK** to accept the default name **Diagram1**.
7. The **Save** dialog box appears with a message that `Table1` will be saved to the database. Click **Yes**.

Modifying Table Structure

You can add check constraints and make relationships between tables in Diagram Designer.

To create check constraints

1. In `Table1`, right-click the `T1col3` row and choose **Check Constraints**.

   The **Check Constraints** dialog box appears.

2. Click **Add**.

   A new constraint appears in the **Selected Check Constraint** list, with the default name `CK_Table1`.

3. Select the **Expression** row in the grid and click the ellipsis button.

   The **Check Constraint Expression** dialog box appears.

4. Type **T1col3 > 5** and click **OK**.

   `Table1` now has a constraint that all values entered into `T1col3` must be greater than 5.

5. Click **Close**.

To create relationships between tables

1. Create a new table in Diagram designer named `Table2` with the following columns:

Expand table

Column Name	Data Type	Allow Nulls
T2col1	**int**	not checked

Column Name	Data Type	Allow Nulls
T2col2	**varchar(50)**	checked
T2col3	**xml**	checked

> ⓘ Note
>
> The columns on the primary key side of a foreign key relationship must participate in either a Primary Key or a Unique Constraint.

2. Drag `T2col1` to `T1col1`.

   Two dialog boxes appear: **Foreign Key Relationship** in the background and **Tables and Columns** in the foreground.

3. Click **OK** to save the new relationship.

4. Click **OK** again.

Creating Indexes

You can create indexes on most types of data, including XML.

To create a standard index

1. Right-click `Table1` and choose **Indexes/Keys**.

   The **Indexes/Keys** dialog box appears.

2. Click **Add**.

   A new index appears in the **Selected Primary/Unique Key or Index** list, with a default name similar to `IX_Table1`.

3. Select the **Columns** row and click the ellipsis button.

   The **Index Columns** dialog box appears.

4. Click the drop-down arrow under **Column Name** and select `T1col2`.

> ⓘ Note

You may add additional columns to this index by selecting the cell below `T1col2` and choosing another column name.

5. Click **OK** to save this index.
6. Click **Close** in the **Indexes/Keys** dialog box.

To create an XML index

1. Right-click `T2col1` and choose **Set Primary Key**.

   > ⓘ **Note**
   >
   > Adding an XML index requires that another column in the table be set as a clustered primary key.

2. Right-click the `T2col3` row in `Table2` and select **XML Indexes**.

   The **XML Indexes** dialog box appears.

3. Click **Add**.

   An XML index with default values will be added to the **Selected XML Index** list.

4. Click **Close**.

   > ⓘ **Note**
   >
   > XML indexes are created per-column. The first XML index is primary; any additional indexes are secondary.

Saving the Diagram

All of the changes you make to a diagram are not posted to the database until you save it. If there are problems or conflicts, a dialog box appears with more information.

To save a database diagram

1. On the **File** menu, select **Save Diagram1**.

The **Save** dialog box appears. If **Warn about Tables Affected** is selected, information about new or changed tables is listed.

2. Click **OK**.

3. If any errors occurred, the **Post-Save Notifications** dialog box appears with the errors and their causes. Fix the errors and save the diagram again.

Next Steps

This is a basic diagram with just two existing and two new tables, but it illustrates the potential for diagramming an existing database or creating a new schema visually. Suggestions for more exploration include:

- Create new diagrams containing groups of related tables
- Customize the amount of information shown for each table
- Change the layout and add annotations
- Copy the diagram to a bitmap

See Also

Customize the Amount of Information Displayed in Diagrams (Visual Database Tools)
Set Up Database Diagram Designer (Visual Database Tools)
Add Tables to Diagrams (Visual Database Tools)
Create Relationships Between Tables on a Diagram (Visual Database Tools)
Create XML Indexes
Copy an Image of a Database Diagram to the Clipboard (Visual Database Tools)
Work with Diagram Layout (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Table-Valued Object (Column) Properties (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

These properties appear when you select a column in a table-valued object in the **Diagram** pane of Query and View Designer.

> **Note**
>
> The properties in this topic are ordered by category rather than alphabet.

> **Note**
>
> The dialog boxes and menu commands you see might differ from those described in Help depending on your active settings or edition. To change your settings, choose **Import and Export Settings** on the **Tools** menu.

Identity Category
Expands to show the **Name** property.

Name
Shows the name of the selected column.

Query DesignerCategory
Expands to show properties for **Allow Nulls**, **Collation**, **Datatype**, **Length**, **Precision**, **Scale**, and **Size**.

Allow Nulls
Shows whether or not the column's data type allows null values.

Collation
Shows the collation setting for the selected column. The collation can be set in the Column Properties tab of Table Designer.

Data Type
Shows the data type of the selected column.

Length
Shows the number of characters or digits allowed by the selected column's data type.

This property is only available for character-based data types.

> **Note**
>
> For the size in bytes, see the **Size** property below.

Precision
Shows the maximum number of digits allowed for numeric data types. This property shows **0** for nonnumeric data types.

Scale
Shows the maximum number of digits that can appear to the right of the decimal point for numeric data types. This value must be less than or equal to the precision. This property shows **0** for nonnumeric data types.

Size
Shows the size in bytes allowed by the column's data type. For example, a nchar data type may have a length of 10 (the number of characters) but it would have a size of 20 to account for Unicode character sets.

Feedback

Was this page helpful? Yes No

How the Query and View Designer Represents Joins (Visual Database Tools)

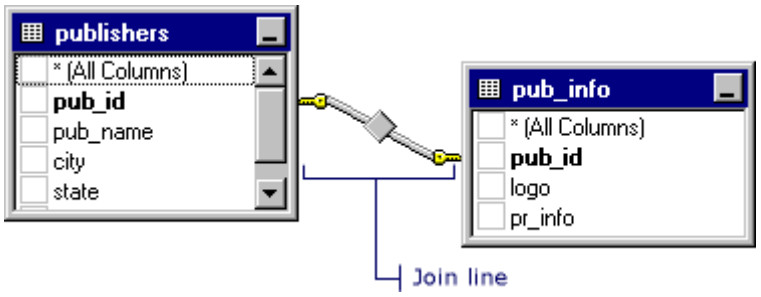
Article • 02/13/2025

Applies to: SQL Server

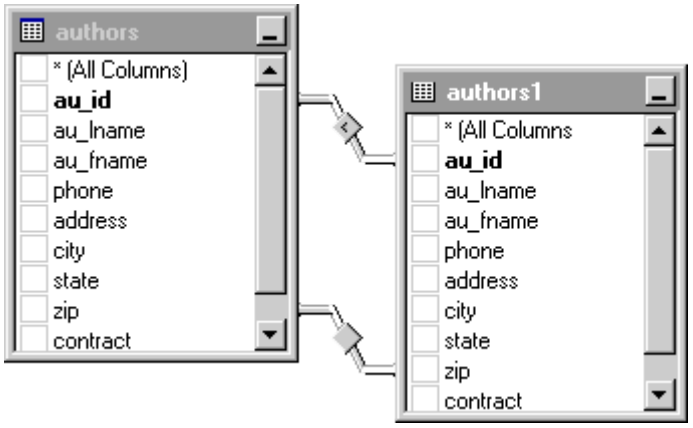
If tables are joined, the Query and View Designer represents the join graphically in the Diagram pane and by using SQL syntax in the SQL pane.

Diagram Pane

In the Diagram pane the Query and View Designer displays a join line between the data columns involved in the join. The Query and View Designer displays one join line for each join condition. For example, the following illustration shows a join line between two tables that are joined:



If tables are joined using more than one join condition, the Query and View Designer displays multiple join lines, as in the following example:



If the joined data columns are not displayed (for example, the rectangle representing the table or table-structured object is minimized or the join involves an expression), the Query and View Designer places the join line at the title bar of the rectangle representing the table or table-structured object.

The shape of the icon in the middle of the join line indicates how the tables or table-structured objects are joined. If the join clause uses an operator other than equal (=), the operator appears in the join line icon. The following table lists the icons that appear in the join line.

Expand table

Join line icon	Description
	Inner join (created using an equal sign).
	Inner join based on the "greater than" operator.
	Outer join in which all rows from the table represented on the left will be included, even if they do not have matches in the related table.
	Outer join in which all rows from the table represented on the right will be included, even if they do not have matches in the related table.
	Full outer join in which all rows from both tables will be included, even if they do not have matches in the related table.

The symbols on the ends of the join line indicate the type of join. The following table lists the types of joins and the icons displayed on the ends of the join line.

Expand table

Icon on ends of join line	Type of join
	One-to-one join.
	One-to-many join.
	Query and View Designer cannot determine the join type. This situation occurs most often when you have created a join manually.

SQL Pane

A join can be expressed in a number of ways in a SQL statement. The exact syntax depends on the database you are using and on how you have defined the join.

Syntax options for joining tables include:

- **JOIN qualifier for the FROM clause**. The keywords INNER and OUTER specify the join type. This syntax is standard for ANSI 92 SQL.

  For example, if you join the `publishers` and `pub_info` tables based on the `pub_id` column in each table, the resulting SQL statement might look like this:

  ```
  SELECT *
  FROM publishers INNER JOIN pub_info ON
     publishers.pub_id = pub_info.pub_id
  ```

  If you create an outer join, the words LEFT OUTER or RIGHT OUTER appear in place of the word INNER.

- **WHERE clause compares columns in both tables**. A WHERE clause appears if the database does not support the JOIN syntax (or if you entered it yourself). If the join is created in the WHERE clause, both table names appear in the FROM clause.

  For example, the following statement joins the `publishers` and `pub_info` tables.

  ```
  SELECT *
  FROM publishers, pub_info
  WHERE publishers.pub_id = pub_info.pub_id
  ```

See Also

Query with Joins (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Design Queries and Views How-to Topics (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

The Query and View Designer can help you create and maintain the data retrieval and data manipulation portions of your application.

In This Section

Open the Query and View Designer (Visual Database Tools)
Provides steps for opening a new or existing query or view in the Query and View Designer.

Navigate in the Query and View Designer (Visual Database Tools)
Provides a list of key combinations and mouse operations you can use in the Query and View Designer.

Query and View Designer Tools (Visual Database Tools)
Provides links to topics describing the various components of the Query and View Designer.

Types of Queries (Visual Database Tools)
Lists procedural topics on how to create various types of queries, such as UPDATE, DELETE, and MAKE TABLE queries.

Work with Data in the Results Pane (Visual Database Tools)
Describes what you can do with the results returned by a query or view and provides links to procedural topics.

Perform Basic Operations with Queries (Visual Database Tools)
Lists topics on creating, modifying, running, and copying queries as well as basic topics on sorting, ordering, and printing query results.

Specify Search Criteria (Visual Database Tools)
Lists rules and conventions for creating queries, as well as procedures on how to use criteria to get specific results.

Query with Joins (Visual Database Tools)
Lists topics on using joins to query multiple tables, such as how to create outer joins and

how to join tables automatically.

Query with Parameters (Visual Database Tools)
Lists topics for creating queries with parameters, which allow a different value to be used each time you run the query.

Related Sections

Design Tables (Visual Database Tools)
Detailed steps for designing and maintaining database structure through tables, columns, keys, indexes, stored procedures, and more.

Feedback

Was this page helpful? Yes No

Use HAVING and WHERE clauses in the same query (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

In some instances, you might want to exclude individual rows from groups (using a `WHERE` clause) before applying a condition to groups as a whole (using a `HAVING` clause).

A `HAVING` clause is like a `WHERE` clause, but applies only to groups as a whole (that is, to the rows in the result set representing groups), whereas the `WHERE` clause applies to individual rows. A query can contain both a `WHERE` clause and a `HAVING` clause. In that case:

- The `WHERE` clause is applied first to the individual rows in the tables or table-valued objects in the **Diagram** pane. Only the rows that meet the conditions in the `WHERE` clause are grouped.
- The `HAVING` clause is then applied to the rows in the result set. Only the groups that meet the `HAVING` conditions appear in the query output. You can apply a `HAVING` clause only to columns that also appear in the GROUP BY clause or in an aggregate function.

Specify a WHERE and HAVING clause on two joined tables

> **Note**
>
> The database used in this article is the `pubs` database, available from Northwind and pubs sample databases for Microsoft SQL Server on GitHub.

For example, imagine that you're joining the `titles` and `publishers` tables to create a query showing the average book price for a set of publishers. You want to see the average price for only a specific set of publishers - perhaps only the publishers in the state of California. And even then, you want to see the average price only if it's over $10.00.

You can establish the first condition by including a `WHERE` clause, which discards any publishers that aren't in California, before calculating average prices. The second condition requires a `HAVING` clause, because the condition is based on the results of grouping and summarizing the data. The resulting SQL statement might look like this:

```
SQL

SELECT titles.pub_id, AVG(titles.price)
FROM titles INNER JOIN publishers
   ON titles.pub_id = publishers.pub_id
WHERE publishers.state = 'CA'
GROUP BY titles.pub_id
HAVING AVG(price) > 10;
```

In the Visual Database Tools of SQL Server Management Studio, you can create both `HAVING` and `WHERE` clauses in the **Criteria** pane. By default, if you specify a search condition for a column, the condition becomes part of the `HAVING` clause. However, you can change the condition to be a `WHERE` clause.

You can create a `WHERE` clause and `HAVING` clause involving the same column. To do so, you must add the column twice to the **Criteria** pane, then specify one instance as part of the `HAVING` clause and the other instance as part of the `WHERE` clause.

Specify a WHERE condition in an aggregate query

1. Specify the groups for your query. For details, see Group Rows in Query Results (Visual Database Tools).

2. If it's not already in the **Criteria** pane, add the column on which you want to base the `WHERE` condition.

3. Clear the **Output** column unless the data column is part of the GROUP BY clause or included in an aggregate function.

4. In the **Filter** column, specify the `WHERE` condition. The Query and View Designer adds the condition to the `HAVING` clause of the SQL statement.

> ⓘ Note
>
> The query shown in the example for this procedure joins two tables, `titles` and `publishers`.

At this point in the query, the SQL statement contains a `HAVING` clause:

```
SQL

SELECT titles.pub_id, AVG(titles.price)
FROM titles INNER JOIN publishers
    ON titles.pub_id = publishers.pub_id
GROUP BY titles.pub_id
HAVING publishers.state = 'CA'
```

5. In the **Group By** column, select **Where** from the list of group and summary options. The Query and View Designer removes the condition from the `HAVING` clause in the SQL statement and adds it to the `WHERE` clause.

   The SQL statement changes to include a `WHERE` clause instead:

```
SQL

SELECT titles.pub_id, AVG(titles.price)
FROM titles INNER JOIN publishers
    ON titles.pub_id = publishers.pub_id
WHERE publishers.state = 'CA'
GROUP BY titles.pub_id;
```

Related content

- SELECT - HAVING (Transact-SQL)
- Sort and Group Query Results (Visual Database Tools)
- Summarize Query Results (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Navigate in the Query and View Designer (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can work in the Query and View Designer using the keyboard or the mouse. Refer to the following tables for specific methods.

Any Pane

Expand table

To	Press	Click
Move among the Query and View Designer panes	F6, SHIFT+F6	Anywhere in the target pane

> **Tip**
>
> You can use the shortcut keys CTRL+TAB to change focus to other panes in addition to the Query and View Designer panes.

Diagram Pane

Expand table

To	Press	Click
Move among tables, other table-structured objects, (and to join lines, if available)	TAB, or SHIFT+TAB	The table, table-structured object, or join line to move to
Move between columns in a table or table-structured object	Arrow keys	The column to go to
Choose the selected data column for output	SPACEBAR or PLUS key	The check box next to the name of the column
Remove the selected data column from the query output	SPACEBAR or MINUS key	The check box next to the name of the column

To	Press	Click
Remove the selected table, table-structured object, or join line from the query	DELETE	Right-click, and then choose **Remove**

> **Note**
>
> If multiple items are selected, pressing this key affects all selected items. Select multiple items by holding down the CTRL key while clicking them.

For more information, see Diagram Pane (Visual Database Tools).

Criteria Pane

Expand table

To	Press	Click
Move among cells	Arrow keys or TAB or SHIFT+TAB	The target cell
Move to the last row in a selected column	CTRL+DOWN ARROW	
Move to the first row in a selected column	CTRL+UP ARROW	
Move to the top left cell in the visible portion of grid	CTRL+HOME	
Move to the bottom right cell	CTRL+END	
Move in a drop-down list	UP ARROW or DOWN ARROW	The button in the cell
Select an entire grid column	CTRL+SPACEBAR	The column header
Toggle between edit mode and cell selection mode	F2	
Copy selected text in cell to the Clipboard (in edit mode)	CTRL+C	
Cut selected text in cell and place it on the Clipboard (in edit mode)	CTRL+X	
Paste text from the Clipboard (in edit mode)	CTRL+V	

To	Press	Click
Toggle between insert and overstrike mode while editing in a cell	INS	
Toggle the check box in the Output column	SPACEBAR	The check box
Clear the selected contents of a cell	DELETE	
Clear all values for a selected grid column	DELETE	
Insert row between existing rows	INS after you select grid row	
Add an Or ... column	INS after you select any Or ... column	

> **Note**
>
> If multiple items are selected, pressing this key affects all selected items.

For more information, see Criteria Pane (Visual Database Tools).

SQL Pane

You can use the standard Windows editing keys when working in the SQL pane, such as CTRL+Arrow keys to move between words, and the **Cut**, **Copy**, and **Paste** commands on the **Edit** menu.

> **Note**
>
> You can only insert text; there is no overstrike mode.

For more information, see SQL Pane (Visual Database Tools).

Results Pane

Expand table

To	Press	Click
Move between cells	Arrow keys or TAB or SHIFT+TAB	The target cell

To	Press	Click
Move to first or last cell in current row	HOME or END	
Move to the first row in the current column	CTRL+UP ARROW	
Move to the top left cell	CTRL+HOME	
Move to the bottom cell in the first column	CTRL+DOWN ARROW	
Select to first character in a cell	SHIFT+HOME	
Select to last character in a cell	SHIFT+END	
Toggle between edit mode and cell selection mode	F2	
Toggle between insert and overstrike mode while editing in a cell	INS	
Delete a row from the table	DELETE	
Undo changes for the current cell	ESC in cell that has changed	
Undo changes for the current row	ESC in any cell that has not changed	
Enter null into a cell	CTRL+0	
Copy selected columns or rows to the Clipboard	CTRL+C	
Copy selected text in cell to the Clipboard (in edit mode)	CTRL+C	
Cut selected text in cell to the Clipboard (in edit mode)	CTRL+X	
Paste text from the Clipboard (in edit mode)	CTRL+V	

> **Note**
>
> If multiple items are selected, pressing this key affects all selected items.

For more information, see Results Pane (Visual Database Tools).

See Also

Design Queries and Views How-to Topics (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Work with Columns in Aggregate Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

When you create aggregate queries the Query and View Designer makes certain assumptions so that it can construct a valid query. For example, if you are creating an aggregate query and mark a data column for output, the Query and View Designer automatically makes the column part of the GROUP BY clause so that you do not inadvertently attempt to display the contents of an individual row in a summary.

Using Group By

The Query and View Designer uses the following guidelines for working with columns:

- When you choose the Group By option or add an aggregate function to a query, all columns marked for output or used for sorting are automatically added to the GROUP BY clause. Columns are not automatically added to the GROUP BY clause if they are already part of an aggregate function.

  If you do not want a particular column to be part of the GROUP BY clause, you must manually change it by selecting a different option in the Group By column of the Criteria pane. However, the Query and View Designer will not prevent you from choosing an option that can result in a query that will not run.

- If you manually add a query output column to an aggregate function in either the Criteria or SQL pane, the Query and View Designer does not automatically remove other output columns from the query. Therefore, you must remove the remaining columns from the query output or make them part of the GROUP BY clause or of an aggregate function.

When you enter a search condition into the Filter column of the Criteria pane, the Query and View Designer follows these rules:

- If the **Group By** column of the grid is not displayed (because you have not yet specified an aggregate query), the search condition is placed into the WHERE clause.

- If you are already in an aggregate query and have selected the option **Where** in the **Group By** column, the search condition is placed into the WHERE clause.

- If the **Group By** column contains any value other than **Where**, the search condition is placed in the HAVING clause.

Using the HAVING and WHERE Clauses

The following principles describe how you can reference columns in an aggregate query in search conditions. In general, you can use a column in a search condition to filter the rows that should be summarized (a WHERE clause) or to determine which grouped results appear in the final output (a HAVING clause).

- Individual data columns can appear in either the WHERE or HAVING clause, depending on how they are used elsewhere in the query.
- WHERE clauses are used to select a subset of rows for summarizing and grouping and are thus applied before any grouping is done. Therefore, you can use a data column in a WHERE clause even if it is not part of the GROUP BY clause or contained in an aggregate function. For example, the following statement selects all titles that cost more than $10.00 and averages the price:

```
SELECT AVG(price)
FROM titles
WHERE price > 10
```

- If you create a search condition that involves a column also used in a GROUP BY clause or aggregate function, the search condition can appear as either a WHERE clause or a HAVING clause - you can decide which when you create the condition. For example, the following statement creates an average price for the titles for each publisher, then displays the average for the publishers in which the average price is greater than $10.00:

```
SELECT pub_id, AVG(price)
FROM titles
GROUP BY pub_id
HAVING (AVG(price) > 10)
```

- If you use an aggregate function in a search condition, the condition involves a summary and must therefore be part of the HAVING clause.

See Also

Summarize Query Results (Visual Database Tools)
Sort and Group Query Results (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Select Rows That Do Not Match a Value (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

To find rows that do not match a value, use the NOT operator.

To find rows that do not match a value

1. If you have not done so already, add the columns or expressions that you want to use within your search condition to the Criteria pane.
2. Locate the row containing the data column or expression to search, and then in the **Filter** grid column, enter the NOT operator followed by a search value.

For example, to find all the rows in a `products` table where the values in the product code column begin with a character other than "A," you can enter a search condition such as the following:

```
NOT LIKE 'A%'
```

See Also

Rules for Entering Search Values
Specify Search Criteria

Feedback

Was this page helpful? Yes No

Save Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

Save your query when you want to work with it later. Note that running your query does not save your changes.

To save a query

1. Create your query.
2. From the **File** menu, click **Save <query_name>**.

   > **Note**
   >
   > A newly created query will be given a system name. To choose your own name for the query choose **Save<query_name> As**.

3. The query will be saved in the Query folder of the project in Solution Explorer.

See Also

Create Queries
Run Queries
Discard Changes Made to Queries
Design Queries and Views How-to Topics
Types of Queries
Perform Basic Operations with Queries

Feedback

Was this page helpful? Yes No

Specify Search Conditions (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can specify the data rows that appear in your query by specifying search conditions. For example, if you are querying an `employee` table, you can specify that you want to find only the employees who work in a particular region.

You specify search conditions using an expression. Most commonly the expression consists of an operator and a search value. For example, to find employees in a particular sales region, you might specify the following search criterion for the `region` column:

```
='UK'
```

> **Note**
>
> If you are working with multiple tables, the Query and View Designer examines each search condition to determine whether the comparison you are making results in a join. If so, the Query and View Designer automatically converts the search condition into a join. For more information, see Join Tables Automatically (Visual Database Tools).

To specify search conditions

1. If you have not done so already, add the columns or expressions that you want to use within your search condition to the Criteria pane.

   If you are creating a Select query and do not want the search columns or expressions to appear in the query output, clear the **Output** column for each search column or expression to remove them as output columns.

2. Locate the row containing the data column or expression to search, and then in the **Filter** column, enter a search condition.

> **Note**
>
> If you do not enter an operator, the Query and View Designer automatically inserts the equality operator "=".

The Query and View Designer updates the SQL statement in the SQL Pane by adding or modifying the WHERE clause.

See Also

Rules for Entering Search Values (Visual Database Tools)
Specify Search Criteria (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Create Outer Joins (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

By default, the Query and View Designer creates an inner join between tables. Inner joins eliminate the rows that do not match with a row from the other table. Outer joins, however, return all rows from at least one of the tables or views mentioned in the FROM clause, as long as those rows meet any WHERE or HAVING search conditions. If you want to include data rows in the result set that do not have a match in the joined table, you can create an outer join.

When you create an outer join, the order in which tables appear in the SQL statement (as reflected in the SQL pane) is significant. The first table you add becomes the "left" table and the second table becomes the "right" table. (The actual order in which the tables appear in the Diagram pane is not significant.) When you specify a left or right outer join, you are referring to the order in which the tables were added to the query and to the order in which they appear in the SQL statement in the SQL pane.

To create an outer join

1. Create the join, either automatically or manually. For details, see Join Tables Automatically (Visual Database Tools) or Join Tables Manually (Visual Database Tools).
2. Select the join line in the Diagram pane, and then from the **Query Designer** menu, choose **Select All Rows from <tablename>**, selecting the command that includes the table whose extra rows you want to include.
    - Choose the first table to create a left outer join.
    - Choose the second table to create a right outer join.
    - Choose both tables to create a full outer join.

When you specify an outer join, the Query and View Designer modifies the join line to indicate an outer join.

In addition, the Query and View Designer modifies the SQL statement in the SQL pane to reflect the change in join type, as shown in the following statement:

```
SELECT employee.job_id, employee.emp_id,
   employee.fname, employee.minit, jobs.job_desc
FROM employee LEFT OUTER JOIN jobs ON
    employee.job_id = jobs.job_id
```

Because an outer join includes unmatched rows, you can use it to find rows that violate foreign key constraints. To do so, you create an outer join and then add a search condition to find rows in which the primary key column of the rightmost table is null. For example, the following outer join finds rows in the `employee` table that do not have corresponding rows in the `jobs` table:

```
SELECT employee.emp_id, employee.job_id
FROM employee LEFT OUTER JOIN jobs
   ON employee.job_id = jobs.job_id
WHERE (jobs.job_id IS NULL)
```

See Also

Query with Joins (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Create Update Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can change the contents of multiple rows in one operation by using an Update query. For example, in a `titles` table you can use an Update query to add 10% to the price of all books for a particular publisher.

When you create an Update query, you specify:

- The table to update.
- The columns whose contents you want to update.
- The value or expression to use to update the individual columns.
- Search conditions to define the rows you want to update.

For example, the following query updates the `titles` table by adding 10% to the price of all titles for one publisher:

```
UPDATE titles
SET price = price * 1.1
WHERE (pub_id = '0766')
```

> **⊗ Caution**
>
> You cannot undo the action of executing an Update query. As a precaution, back up your data before executing the query.

To create an Update query

1. Add the table you want to update to the Diagram pane.
2. From the **Query Designer** menu point to **Change Type**, and then click **Update**.

> **ⓘ Note**

If more than one table is displayed in the Diagram pane when you start the Update query, the Query and View Designer displays the Choose Target Table for Insert Values dialog box to prompt you for the name of the table to update.

3. In the Diagram pane, click the check box for each column for which you want to supply new values. Those columns will show in the Criteria pane. Columns will be updated only if you add them to the query.

4. In the **New Value** column of the Criteria pane, enter the update value for the column. You can enter literal values, column names, or expressions. The value must match (or be compatible with) the data type of the column you are updating.

   > ⊗ **Caution**
   >
   > The Query and View Designer cannot check that a value fits within the length of the column you are updating. If you provide a value that is too long, it might be truncated without warning. For example, if a `name` column is 20 characters long but you specify an update value of 25 characters, the last 5 characters might be truncated.

5. Define the rows to update by entering search conditions in the **Filter** column. For details, see Specify Search Criteria (Visual Database Tools).

   If you do not specify a search condition, all rows in the specified table will be updated.

   > ⓘ **Note**
   >
   > When you add a column to the Criteria pane for use in a search condition, the Query and View Designer also adds it to the list of columns to be updated. If you want to use a column for a search condition but not update it, clear the check box next to the column name in the rectangle representing the table or table-valued object.

When you execute an Update query, no results are reported in the Results Pane. Instead, a message appears indicating how many rows were changed.

See Also

Supported Query Types (Visual Database Tools)
Design Queries and Views How-to Topics (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Copy Tables from One Database Diagrams to Another (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

You can copy a table from one database diagram to another in the same database.

Copying a table from one database diagram to another diagram adds a reference to the table in the second diagram. The table is not duplicated in your database. For example, if you copy the `authors` table from one database diagram to another, each diagram references the same `authors` table in the database.

To copy a table from another database diagram

1. Make sure you are connected to the database whose table you want to copy.
2. Open the source and target database diagrams and within the source diagram, select the table that you want to copy to the target diagram.
3. Click the **Copy** button on the toolbar. This action places the selected table definition on the Clipboard.
4. Switch to the target diagram. This diagram must be in the same database as the source diagram.
5. Click the **Paste** button on the toolbar. The Clipboard contents appear at the new location and remain highlighted until you click elsewhere. If relationships exist between the selected tables and other tables in the target diagram, relationship lines are automatically drawn.

When you edit the table in either diagram, your changes are reflected in both diagrams. Similarly, once you save the table in either diagram, the table is no longer considered "modified" in either diagram.

See Also

Work with Database Diagrams (Visual Database Tools)
Add Tables to Diagrams (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Change the Font Color, Size, and Style in Diagrams (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

You can customize the way text appears in database diagrams by changing the font color, size, and style. The changes are applied to new diagrams and to existing diagrams when they are modified.

To change the font color, size, and style in diagrams

1. On the **Tools** menu, click **Options**.
2. Expand **Environment**, and then click **Fonts and Colors**.
3. In the **Show settings for** list, select **Table and Database Designers**.
4. Change the font, size, display item, foreground and background colors.

See Also

Work with Database Diagrams (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Verify Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

To avoid problems, you can check the query you have built to ensure its syntax is correct. This option is especially useful when you enter statements in the SQL pane.

Some notes to keep in mind about verifying queries:

- A statement can be valid, and therefore be verified successfully, even if it cannot be represented in the **Diagram Pane** and **Criteria Pane**.
- SQL Verification can detect some, but not all SQL errors. If a query contains an error not detected during SQL verification, the database will detect the error when you run the query.
- Queries that contain parameters cannot be verified.

To verify a SQL statement

- Right-click in the **SQL Pane**, and select **Verify SQL Syntax** from the shortcut menu.

See Also

Run Queries (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

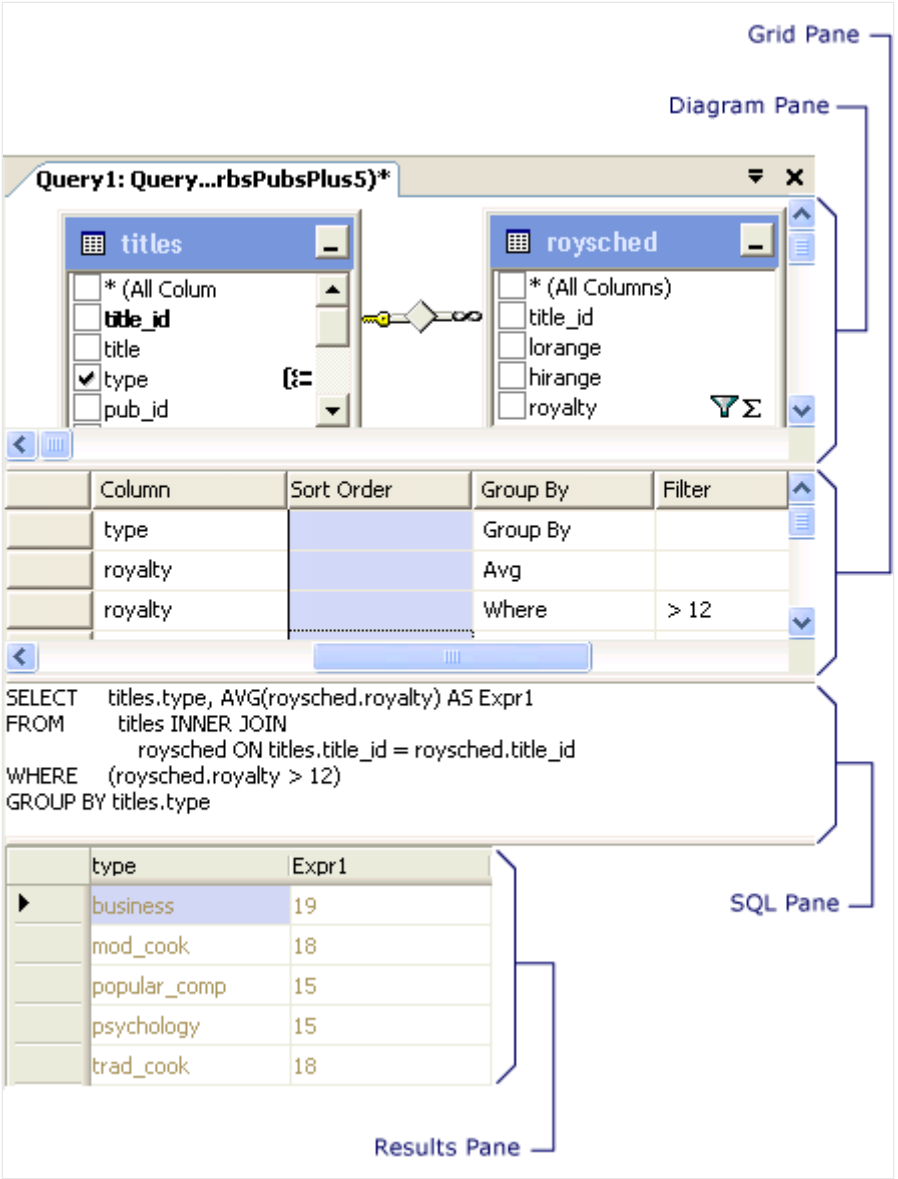
Was this page helpful? Yes No

Query and View Designer Tools (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

When you design a query, view, in-line function, or single-statement stored procedure, the designer you use consists of four panes: the Diagram pane, the Criteria pane, the SQL pane, and the Results pane.



- The Diagram pane displays the tables and other table-valued objects that you're querying. Each rectangle represents a table or table-valued object and shows the available data columns. Joins are indicated by lines between the rectangles. For more information, see Diagram Pane (Visual Database Tools).

- The Criteria pane contains a spreadsheet-like grid in which you specify options, such as which data columns to display, what rows to select, how to group rows, and so on. For more information, see Criteria Pane (Visual Database Tools).
- The SQL pane displays the SQL statement for the query or view. You can edit the SQL statement created by the Designer or you can enter your own SQL statement. It's particularly useful for entering SQL statements that can't be created using the Diagram and Criteria panes, such as union queries. For more information, see SQL Pane (Visual Database Tools).
- The Results pane shows a grid with data retrieved by the query or view. In the Query and View Designer, the pane shows the results of the most recently executed SELECT query. You can modify the database by editing values in the cells of the grid, and you can add or delete rows. For more information, see Results Pane (Visual Database Tools).

You can create a query or view by working in any of the panes: you can specify a column to display by choosing it in the Diagram pane, entering it into the Criteria pane, or making it part of the SQL statement in the SQL pane.

Displaying and hiding panes

To hide a pane or display a pane that isn't visible, right-click the design surface, point to **Pane**, and then select the name of the pane. If the Query and View Designer is opened in Query Designer mode, the **Results** pane isn't available.

Related content

- Design Queries and Views How-to articles (Visual Database Tools)
- Open the Query and View Designer (Visual Database Tools)
- SQL Editor (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Remove Tables from Database Diagrams (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

You can remove a table from your database diagram. Removing a table does not alter your database. The table and its relationships to other tables continue to exist in the database.

To remove a table from a database diagram

1. In your database diagram, select the table you want to remove.
2. Right-click the table and choose **Remove Table from Diagram** from the shortcut menu.

   -or-

   Press the ESC key.

   If the table has unsaved changes as a result of edits you made in the database diagram, a message prompts you to save the table before removing it.

The table is removed from your diagram but it continues to exist in the database.

See Also

Work with Database Diagrams (Visual Database Tools)
[How to: Delete Tables from a Database(https://msdn.microsoft.com/ca6aa3e9-9885-44c3-bafc-aec441fd97ec)

Feedback

Was this page helpful? Yes No

Add Text Annotations to Diagrams (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

To make notes or comments on a diagram you can add text annotations.

To add text annotations to diagrams

1. Open the diagram to which you want to add an annotation.
2. Right-click in the empty space of the Diagram window.
3. From the shortcut menu click **New Text Annotation**.
4. Type your note into the text field that appears.

> **Note**
>
> To change the font, right-click the text field and click **Set Text Annotation Font**.

See Also

Work with Database Diagrams (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Create Delete Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can delete all rows in a table by using a Delete query.

> **Note**
>
> Deleting all rows from a table clears the data in the table but does not delete the table itself. To delete a table from a database, right-click the table in Object Explorer and click **Delete**.

When you create a Delete query, the Criteria pane changes to reflect the options available for deleting rows. Because you do not display data in a Delete query, the Output, Sort By, and Sort Order columns are removed. In addition, the check boxes next to the column names in the rectangle representing the table or table-valued object are removed because you cannot specify individual columns to delete.

If the Query and View Designer can't delete one or more of the rows none of them will be deleted and you will receive a message telling you which row(s) contain information that can't be deleted from the database.

> **Caution**
>
> You cannot undo the action of executing a Delete query. As a precaution, back up your data before executing a Delete query.

To create a Delete query

1. Add the table to delete rows from to the Diagram pane.
2. From the **Query Designer** menu, point to **Change Type**, and then click **Delete**.
   Note If more than one table is displayed in the Diagram pane when you start the Delete query, the Query and View Designer displays the Delete Table dialog box to prompt you for the name of the table to delete rows from.

When you execute the Delete query, no results are reported in the Results Pane. Instead, a message appears indicating how many rows were deleted.

See Also

Supported Query Types (Visual Database Tools)
Design Queries and Views How-to Topics (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Delete Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can delete a query, removing it from the project and permanently deleting it from the file system. Queries deleted using the following procedure are not sent to the Recycle Bin and cannot be recovered.

To permanently delete a query

1. From the **View** menu, choose Solution Explorer.
2. In Solution Explorer, select the query you want to permanently delete.
3. Right-click the query and from the shortcut menu, click **Remove**.
4. In the **Microsoft SQL Server Management Studio** dialog box, click **Delete**.

> **Note**
>
> If you want to remove the query from the project without deleting it, click **Remove**. Then you can right-click the query folder and choose **Add Item** to find and add the query back later.

See Also

Clear Query Results (Visual Database Tools)
Discard Changes Made to Queries (Visual Database Tools)
Design Queries and Views How-to Topics (Visual Database Tools)
Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Query Properties (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

These properties appear in the Properties window when you have a query open in Query and View Designer. Unless otherwise noted, you can edit these properties in the Properties window.

> **Note**
>
> The properties in this topic are ordered by category, rather than alphabetically.

Options

Identity Category
Expand to show the **Name** property.

Name
Shows the name of the current query. Cannot be changed in Management Studio.

Database Name
Shows the name of the data source for the selected table.

Server Name
Shows the name of the server for the data source.

Query Designer Category
Expands to show the remaining properties.

Destination table
Specify the name of the table into which you are inserting data. This list appears if you are creating an INSERT query or MAKE TABLE query. For an INSERT query, select a table name from the list.

For a MAKE TABLE query, type the name of the new table. To create a destination table in another data source, specify a fully qualified table name, including the name of the target data source, the owner (if required), and the name of the table.

> **Note**
>
> Query Designer does not check whether the name is already in use or whether you have permission to create the table.

Distinct values
Specify that the query will filter out duplicates in the result set. This option is useful when you are using only some of the columns from the table or tables and those columns might contain duplicate values, or when the process of joining two or more tables produces duplicate rows in the result set. Choosing this option is equivalent to inserting the word DISTINCT into the statement in the SQL pane.

GROUP BY Extension
Specify that additional options for queries based on aggregate queries are available. (Applies only to SQL Server.)

Output All Columns
Specify that all columns from all tables in the current query will be in the result set. Choosing this option is equivalent to specifying an asterisk (*) in place of individual column names after the SELECT keyword in the SQL statement.

Query Parameter List
Shows query parameters. To edit the parameters click the property and then click the ellipses **(...)** to the right of the property. (Applies only to generic OLE DB.)

SQL Comment
Shows a description of the SQL statements. To see the entire description or to edit it, click the description and then click the ellipses **(...)** to the right of the property. Your comments can include information such as who uses the query and when they use it. (Applies only to SQL Server 7.0 databases or later.)

Top Specification Category
Expand to show properties for the **Top**, **Percent**, **Expression**, and **With Ties** properties.

(Top)
Specify hat the query will include a TOP clause, which returns only the first *n* rows or first *n* percentage of rows in the result set. The default is that the query returns the first 10 rows in the result set.

Use this box to change the number of rows to return or to specify a different percentage. (Applies only to SQL Server or later.)

Expression
Specify the number or percentage of rows that the query will return. If you set **Percent** to Yes, this number is the percentage of rows the query will return; if you set **Percent** to No, it represents the number of rows to return. (Applies only to SQL Server version 7.0 or later.)

Percent
Specify that the query will return only the first *n* percent of rows in the result set. (Applies only to SQL Server version 7.0 or later.)

With Ties
Specify that the view will include a WITH TIES clause. WITH TIES is useful if a view includes an ORDER BY clause and a TOP clause based on percentage. If this option is set, and if the percentage cutoff falls in the middle of a set of rows with identical values in the ORDER BY clause, the view is extended to include all such rows. (Applies only to SQL Server version 7.0 or later.)

See Also

Query with Parameters (Visual Database Tools)
Design Queries and Views How-to Topics (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Open Database Diagrams (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Analytics Platform System (PDW)

You can open database diagrams to view or edit the diagram's structure.

To open a database diagram

1. In Object Explorer, expand the **Database Diagrams** folder.
2. Double-click the name of the database diagram you want to open.

   -or-

   Right-click the name of the database diagram you want to open, and then choose **Design Database Diagram**.

The database diagram is opened in the Database Diagram Designer, from which you can edit the diagram.

> **Note**
>
> Only the owner of the diagram or a member of the db_owner role of the database can open the diagram.

See Also

Work with Database Diagrams (Visual Database Tools)
Understand Database Diagram Ownership (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Specify Multiple Search Conditions for Multiple Columns (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

You can expand or narrow the scope of your query by including several data columns as part of your search condition. For example, you might want to:

- Search for employees who either have worked more than five years at the company or who hold certain jobs.
- Search for a book that is both published by a specific publisher and pertains to cooking.

To create a query that searches for values in either of two (or more) columns, you specify an OR condition. To create a query that must meet all conditions in two (or more) columns, you specify an AND condition.

Specifying an OR Condition

To create multiple conditions linked with OR, you put each separate condition in a different column of the Criteria pane.

To specify an OR condition for two different columns

1. In the Criteria Pane, add the columns you want to search.
2. In the **Filter** column for the first column to search, specify the first condition.
3. In the **Or...** column for the second data column to search, specify the second condition, leaving the **Filter** column blank.

   The Query and View Designer creates a WHERE clause that contains an OR condition such as the following:

```
SELECT job_lvl, hire_date
FROM employee
```

```
WHERE (job_lvl >= 200) OR
  (hire_date < '01/01/1998')
```

4. Repeat Steps 2 and 3 for each additional condition you want to add. Use a different **Or...** column for each new condition.

Specifying an AND Condition

To search different data columns using conditions linked with AND, you put all the conditions in the **Filter** column of the grid.

To specify an AND condition for two different columns

1. In the Criteria Pane, add the columns you want to search.
2. In the **Filter** column for the first data column to search, specify the first condition.
3. In the **Filter** column for the second data column, specify the second condition.

   The Query and View Designer creates a WHERE clause that contains an AND condition such as the following:

```
SELECT pub_id, title
FROM titles
WHERE (pub_id = '0877') AND (title LIKE '%Cook%')
```

4. Repeat Steps 2 and 3 for each additional condition you want to add.

See Also

Combine Conditions When AND Has Precedence
Combine Conditions When OR Has Precedence
Conventions for Combining Search Conditions in the Criteria Pane
Specify Search Criteria

Feedback

Was this page helpful? Yes No

Add Derived Tables to Queries (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

Derived tables are result sets used as table sources in a query. You can add a derived table to a query in the **Diagram Pane**.

To add a derived table to a query

1. Open an existing query or create a new query.
2. Right-click the **Diagram Pane** and choose **Add New Derived Table**.

   A new table with the name derivedtbl_*N* is added, and the derived table's SELECT statement is added to the query's FROM clause.

See Also

Perform Basic Operations with Queries (Visual Database Tools)
Create Queries (Visual Database Tools)
Open Queries (Visual Database Tools)
SELECT (Transact-SQL)

Feedback

Was this page helpful? Yes No

SQL Editor (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

Use the SQL Editor to edit existing stored procedures, functions, triggers, and SQL scripts. This window opens when you open any of those objects. If you want to create a new SQL statement to run against your data source, use the SQL Pane of Query Designer.

The SQL editor provides many useful SQL text-editing features, including:

- Color coding of SQL keywords to minimize syntax and spelling errors.
- Generating skeletal stored procedures and triggers.
- Providing useful editing functions, including cut, copy, paste, and dragging operations.
- Changing the editor's behavior (by selecting **Options** from the **Tools** menu) to modify virtual spaces, word wrap, line numbers, and tab size.
- Helping manage debugging breakpoints.

> **Note**
>
> The SQL Editor does not have IntelliSense prompting.

When editing SQL statements, certain Transact-SQL statements are enclosed in a box surrounded by a thin line. This helps to visually break the SQL code into command sections, and identifies blocks of SQL statements that can be graphically designed using Query Designer. For more information on using Query Designer, see Design Queries and Views How-to Topics (Visual Database Tools).

See Also

Design Queries and Views How-to Topics (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Print Query Results (Visual Database Tools)

Article • 02/13/2025

Applies to: SQL Server

To print the results of your query, you can copy the contents of the Results pane to another Windows program, such as a word processing program, then format and print it there.

The Query and View Designer puts query results onto the Clipboard using tabs as delimiters between columns and carriage return and linefeed characters as delimiters between rows.

To print query results

1. Select the columns or rows that you want to print.
2. Use the **Copy** command to move them to the Clipboard.
3. Switch to the Windows program you want to use to print the results.
4. Use the **Paste** command to move the query results from the Clipboard.
5. Format and print the results.

See Also

Perform Basic Operations with Queries (Visual Database Tools)

Feedback

Was this page helpful? Yes No

Microsoft SQL documentation

Learn how to use SQL Server and Azure SQL, both on-premises and in the cloud.

DOWNLOAD
Try SQL Server 2022

OVERVIEW
Get an Azure VM with SQL Server

OVERVIEW
SQL tools

REFERENCE
Troubleshoot

SQL Server on Windows

On-premises

On an Azure VM

Previous versions (SQL 2005-2014)

SQL on Azure

Azure SQL Database

Azure SQL Managed Instance

Azure SQL Edge

Azure Synapse Analytics

SQL Server on Linux

On-premises

On an Azure VM

Getting started

- What is SQL Server?
- Connect to the Database Engine
- Connect and query with SSMS
- Write Transact-SQL
- SQL Server in a Linux container
- SQL Docs navigation tips

What's new

- SQL Server 2022
- Azure SQL Database
- Azure SQL Managed Instance
- SQL Server on Azure VMs
- Azure Data Studio
- SQL Server Machine Learning Services

Educational SQL resources

SQL Server enabled by Azure Arc

Install

- SQL Server
- SQL Server Management Studio (SSMS)
- Azure Data Studio
- Analysis Services (SSAS)
- Integration Services (SSIS)
- Reporting Services (SSRS)

Migrate

- Data migration
- Database compatibility certification
- Azure Database Migration Service (DMS)
- Database Migration Assistant (DMA)
- SQL Server Migration Assistant (SSMA)

See more >

Development

- SQL Server client connectivity
- Azure SQL Database applications
- Azure SQL Database connectivity libraries
- Transact-SQL (T-SQL)
- SQL PowerShell
- Database samples

Reporting and analysis

- Power BI Report Server
- Power BI Service
- SQL Server Reporting Services (SSRS)
- SQL Server Analysis Services (SSAS)
- Azure Analysis Services

Big data and machine learning

- Azure Synapse Analytics
- SQL Server Machine Learning Services (R & Python)
- SQL Server Big Data Clusters
- Data Virtualization with PolyBase
- Microsoft Analytics Platform System

SQL Server Management Studio (SSMS) support policy

Article • 02/13/2025

The support policy for SQL Server Management Studio is stated as follows.

Support policy for SSMS

- Starting with SSMS 17.0, the SQL Tools team has adopted the Microsoft Modern Lifecycle Policy.
- Read the original Modern Lifecycle Policy announcement.
- For more information, see Modern Policy FAQs.

For the best experience, we recommend that you install the most recent version via https://aka.ms/ssms.

SSMS updates

Beginning with version 18.0, all security updates, critical updates, hotfixes, and any new features, are released only in the latest point release of a major version. Once a new version of SSMS is released to the public, whether it's a point release within a major version or a major version itself, we recommend that you update to the latest release.

For example, SSMS 19.3 was released in January 2024, and SSMS 20.2 was released in July 2024. If a customer is using SSMS 19.3 and encounters an issue, the customer must upgrade to SSMS 20.2 and determine if the issue still exists. If the issue still exists in SSMS 20.2, it should be reported to the SSMS team. A fix may be then provided in a later release of SSMS.

The SSMS team doesn't actively back-port fixes to an earlier release.

Full support servicing phase: When running the latest current branch version of SQL Server Management Studio, you receive all updates, such as security updates, critical updates, and new features.

Refer to the following table for SQL Server Management Studio servicing support.

Expand table

Version	November 2018	June 2022	January 2024	March 2024	April 2024
17.9.1	Full support	Upgrade	Upgrade	Upgrade	Upgrade
18.12.1		Full support	Upgrade	Upgrade	Upgrade
19.3			Full support	Upgrade	Upgrade
20.0				Full support	Upgrade
20.2					Full support

> **Note**
>
> The latest current branch version is always in the **Full support** servicing phase. This support statement means that if you encounter a code defect that warrants a critical update, you must have the latest current branch version installed in order to receive a fix.

Feedback

Was this page helpful? Yes No

User Assistance in SQL Server Management Studio

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

User assistance is available in SQL Server Management Studio through the Help menu and SQL Server Books Online. The Help menu in Management Studio offers several different routes to information about SQL Server. It also provides access to SQL Server community and MSDN Online resources not previously available from within the Help environment. In addition, the Help environment is now configurable to launch either within the SQL Server Management Studio environment or in an associated external window of its own.

The Help Interface

The **Contents** and **Index** provide functionality and an interface already familiar to SQL Server users. The other options are:

- **How Do I**

  Provides a hierarchical set of linked pages containing useful topics related to common SQL Server tasks. The content is organized by component and task, for example, Replication topics, and so on.

- **Search**

  Searches for topics, with or without predefined filters. Search in SQL Server is a separate tabbed page. Users can refine their searches with one or more predefined topic type, language, or technology filters. By default, Search does not use any of the predefined filters, and only topics in the installed collections are searched.

  Users can include online resources in their search by enabling online Help. For more information, see "MSDN Online and SQL Server Communities" later in this topic.

- **Dynamic Help**

  Automatically displays links to relevant information while users work in the Management Studio environment.

- **Help Favorites**

  Stores user topic bookmarks for easy access later.

Help on Help (Microsoft Document Explorer Help) links users to the documentation about the Help Viewer, but the topics are in a collection separate from SQL Server Books Online. For information about the Help Viewer, select **Help on Help** from the Help menu of SQL Server Books Online.

MSDN Online and SQL Server Communities

Help in Management Studio also provides users ways to contact MSDN Online and SQL Server-focused communities on the Web for information. You can:

- Access SQL Server communities from the How Do I page.
- Search MSDN Online and SQL Server community sites.

To access SQL Server-focused communities from the How Do I page

1. In SQL Server Management Studio, on the **Help** menu, click **How Do I**.
2. The SQL Server **How Do I** page opens. In the Community Links sidebar, click the name of the community site you want to access.

   > ⓘ **Note**
   >
   > The computer running SQL Server must have a direct connection to the Web.

   Before you can search MSDN Online or the SQL Server communities, you must enable online search.

To enable online search

1. On the **Tools** menu, click **Options**. In the **Options** dialog box, expand the **Environment** and **Help** nodes if necessary, and then click **Online**.
2. In the **When loading Help content** area, select an online option.
3. In the **Search these providers** list, select the search providers you want to search, and clear those you don't.

4. If **Codezone Community** is one of your selected search providers, then in the **Codezone Community** list, select and clear items as appropriate.

5. Click **OK**.

To search MSDN Online and SQL Server-focused communities from the Search page

1. On the **Help** menu, click **Search**.

2. Enter your search terms in the **Search for** text box, and then click **Search**.

Whether or not you perform a search using the filters available (technology, language, and topic type), your search will now be run against all the search providers you selected.

Launching Help

There are two ways to display Help from SQL Server Management Studio. By default, when SQL Server Books Online is opened from within Management Studio, it opens in a document window external to the Management Studio environment. This window is still associated with the Management Studio; it can respond to some Management Studio events; and when you close Management Studio, Books Online will close as well. Opening Books Online this way is particularly useful when you are using two monitors; you can drag the Books Online window to the second monitor, out of the way of work you are doing in the first one, but still easily referenced.

You can also open Books Online as a document window inside Management Studio. This is preferable when you have limited screen space and want to take advantage of Management Studio and its ability to hide windows.

> ⓘ **Note**
>
> If you want Books Online to be completely independent of Management Studio, open SQL Server Books Online from the **Start** menu, and it will not react to your actions in the Management Studio environment, nor will it close if you exit Management Studio.

To configure Help and SQL Server Books Online to launch inside the Management Studio window

1. On the **Tools** menu, click **Options**, expand **Environment**, expand **Help**, and then click **General**.
2. In the **Show Help Using** box, click **Integrated Help Viewer**.

Feedback

Was this page helpful? Yes No

Local audit for SSMS usage and diagnostic data collection

Article • 02/13/2025

Applies to: SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics Analytics Platform System (PDW)

SQL Server Management Studio (SSMS) contains Internet-enabled features that collect and send anonymous feature usage and diagnostic data to Microsoft.

SSMS might collect standard computer usage and performance information, which could be transmitted to Microsoft and analyzed to improve its quality, security, and reliability.

However, SSMS doesn't collect your name, address, or other data that could identify you as an individual.

For more information, see the Microsoft Privacy Statement and the SQL Server privacy supplement.

Audit feature usage and diagnostic data

To view the feature usage data collected by SSMS, follow these steps:

1. Launch SSMS.
2. Select **View** from the main menu, then select **Output** to display the **Output** window.
3. In the **Output** window, choose **Telemetry** from the **Show output from:** dropdown list menu.

As you interact with SSMS, the **Output** window displays the collected data.

Enable or disable usage and diagnostic data collection in SSMS

To opt in or out of SSMS usage data collection, use the following methods:

- **Registry Subkey:** `HKEY_CURRENT_USER\Software\Microsoft\SQL Server Management Studio\20.0_IsoShell`
- **Registry Entry Name:** `UserFeedbackOptIn`

- **Entry Type:** `DWORD`
- **Values:** `0` to opt out, `1` to opt in

Additionally, SSMS is based on the Visual Studio shell, which enables customer feedback by default. For more information and opt-out options, see Visual Studio Customer Experience Improvement Program.

Registry-based Group Policy on these registry subkeys is honored by SQL Server usage and diagnostic data collection.

Related content

- Configure usage and diagnostic data collection for SQL Server (CEIP)
- Local audit for SQL Server usage and diagnostic data collection (CEIP)

Feedback

Was this page helpful? Yes No